

DODLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

INTERFRAME IMAGE PROCESSING WITH
APPLICATION TO TARGET DETECTION AND TRACKING

by

José Leite Pereira Filho

March 1979

Thesis Advisor:

H. A. Titus

Approved for public release; distribution unlimited.

T187819

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|-----------------------|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) Interframe Image Processing With Application to Target Detection and Tracking | | 5. TYPE OF REPORT & PERIOD COVERED Ph.D. Thesis; March 1979 |
| 7. AUTHOR(s) José Leite Pereira Filho | | 6. PERFORMING ORG. REPORT NUMBER |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 | | 8. CONTRACT OR GRANT NUMBER(s) |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 12. REPORT DATE March 1979 |
| | | 13. NUMBER OF PAGES 251 |
| | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Interframe Image Processing Target Detection and Tracking Two-Dimensional Recursive Filter Three-Dimensional Recursive Filter Hybrid Filter | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The restoration of images and the enhancement and detection of targets in cluttered background are the subjects of this research. The statistical approach is used in order to exploit temporal as well as spatial image redundancies. The images are modeled as a homogeneous random field. An autocorrelation function and a method of parameter identification are proposed. Experiments with several pictures | | |

(20. ABSTRACT Continued)

are presented to validate the model.

An analysis of two-dimensional recursive filters is presented. A three-dimensional recursive filter is developed which exploits the spatial as well as the temporal image redundancies.

A class of hybrid filters is proposed which improves the performance of the recursive filters. Several experiments with pictures are presented to show the ability of the hybrid filters in picture restoration.

A detector is developed for purposes of target extraction from cluttered background images. The detection is independent of the target shape.

A simulation of the target detection and tracking problem is presented. The target is tracked from frame to frame by means of a conventional Kalman filter, which uses the image filter as the measurement device.

Interframe Image Processing with
Application to Target Detection and Tracking

by

José Leite Pereira Filho
Lieutenant Commander, Brazilian Navy
B.S., Pontificia Universidade Católica RJ, 1970
M.S., Naval Postgraduate School, 1977

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

P-3317
1

ABSTRACT

The restoration of images and the enhancement and detection of targets in cluttered background are the subjects of this research. The statistical approach is used in order to exploit temporal as well as spatial image redundancies.

The images are modeled as a homogeneous random field. An autocorrelation function and a method of parameter identification are proposed. Experiments with several pictures are presented to validate the model.

An analysis of two-dimensional recursive filters is presented. A three-dimensional recursive filter is developed which exploits the spatial as well as the temporal image redundancies.

A class of hybrid filters is proposed which improves the performance of the recursive filters. Several experiments with pictures are presented to show the ability of the hybrid filters in picture restoration.

A detector is developed for purposes of target extraction from cluttered background images. The detection is independent of the target shape.

A simulation of the target detection and tracking problem is presented. The target is tracked from frame to frame by means of a conventional Kalman filter, which uses the image filter as the measurement device.

TABLE OF CONTENTS

| | | |
|------|---|----|
| I. | INTRODUCTION ----- | 9 |
| II. | IMAGE MODELING ----- | 15 |
| | A. INTRODUCTION ----- | 15 |
| | B. DYNAMIC MODEL ----- | 15 |
| | C. AN AUTO-REGRESSIVE MODEL ----- | 27 |
| | 1. Autocorrelation Function ----- | 27 |
| | 2. Parameter Identification ----- | 33 |
| | 3. Modeling of Noisy Images ----- | 38 |
| | D. EXPERIMENTAL RESULTS ----- | 45 |
| | 1. Autocorrelation Function ----- | 46 |
| | 2. Parameter Identification ----- | 47 |
| | 3. Bandwidth Compression ----- | 47 |
| III. | TWO-DIMENSIONAL RECURSIVE FILTERS ----- | 62 |
| | A. INTRODUCTION ----- | 62 |
| | B. ALGORITHM FOR COMPUTATION OF ERROR VARIANCE ----- | 64 |
| | 1. Dynamic Model ----- | 64 |
| | 2. Filter Structure ----- | 66 |
| | 3. Variance of the Estimation Error ---- | 68 |
| | a. Variance of the Estimate ----- | 69 |
| | b. Covariance of X and \hat{X} ----- | 71 |
| | 4. Calculation of the Coefficients ---- | 72 |
| | C. ANOTHER RECURSIVE FILTER ----- | 73 |
| | D. NON-RECURSIVE FILTER ----- | 76 |
| | E. PERFORMANCE OF THE FILTERS ----- | 78 |

| | | |
|-----|---|-----|
| IV. | THREE-DIMENSIONAL RECURSIVE FILTER ----- | 87 |
| A. | INTRODUCTION ----- | 87 |
| B. | FILTER DESIGN ----- | 87 |
| | 1. Filter Structure ----- | 90 |
| | 2. Orthogonality Principle ----- | 91 |
| | 3. Gain Computation ----- | 96 |
| C. | PERFORMANCE OF THE FILTER ----- | 97 |
| D. | RESULTS ----- | 100 |
| V. | HYBRID FILTERS ----- | 110 |
| A. | INTRODUCTION ----- | 110 |
| B. | TWO-DIMENSIONAL HYBRID FILTERS ----- | 111 |
| | 1. Two-Dimensional Hybrid Smoother ---- | 111 |
| | a. Performance and Comparison ----- | 117 |
| | b. Experimental Results ----- | 119 |
| | 2. Two-Dimensional Hybrid Predictor --- | 122 |
| C. | THREE-DIMENSIONAL HYBRID FILTERS ----- | 138 |
| | 1. Three-Dimensional Hybrid Smoother -- | 138 |
| | 2. Three-Dimensional Hybrid Predictor - | 140 |
| D. | COMPARISON OF THE PREDICTORS ----- | 143 |
| VI. | TARGET DETECTION ----- | 146 |
| A. | INTRODUCTION ----- | 146 |
| B. | TARGET AND BACKGROUND REGIONS ----- | 147 |
| C. | LIKELIHOOD RATIO TEST ----- | 149 |
| D. | SPECIAL CASES ----- | 162 |
| | 1. Case I: $\alpha = 1, \sigma_T^2 = 0$ ----- | 162 |
| | 2. Case II: $\alpha = 1, \sigma_T^2 \neq 0$ ----- | 167 |

| | |
|---|-----|
| 3. Case III: $\alpha = 0, \sigma_T^2 = 0$ ----- | 175 |
| 4. Case IV: $\alpha = 0, \sigma_T^2 \neq 0$ ----- | 179 |
| VII. TARGET TRACKING ----- | 182 |
| A. INTRODUCTION ----- | 182 |
| B. TARGET TRACKING FILTER ----- | 183 |
| 1. Dynamic Model ----- | 185 |
| 2. The Observation Vector ----- | 187 |
| 3. Kalman Filter ----- | 191 |
| C. FILTER APPLICATION ----- | 199 |
| VIII. COMPUTER SIMULATION ----- | 202 |
| A. DESCRIPTION OF THE SYSTEM ----- | 202 |
| 1. Time-Frame Generation ----- | 204 |
| 2. Image Processing ----- | 205 |
| 3. Target Tracking Filter ----- | 209 |
| B. SELECTED RESULTS ----- | 209 |
| 1. Background Prediction ----- | 209 |
| 2. Transition Region ----- | 211 |
| 3. Background Suppression ----- | 215 |
| 4. Target Tracking ----- | 224 |
| IX. CONCLUSIONS ----- | 243 |
| A. SUMMARY OF RESULTS ----- | 243 |
| B. SUGGESTIONS FOR FURTHER RESEARCH ----- | 247 |
| LIST OF REFERENCES ----- | 249 |
| INITIAL DISTRIBUTION LIST ----- | 251 |

ACKNOWLEDGEMENTS

I am deeply indebted to the Brazilian Navy and the United States Navy for providing me the opportunity to pursue this work.

I wish to express my gratitude to Professor H. A. Titus, my advisor, and to the members of my doctoral committee.

I. INTRODUCTION

A. PROBLEM DEFINITION

This research involves two aspects of digital image processing. One is the restoration of images degraded by white Gaussian observation noise. The other is the enhancement and detection of targets immersed in cluttered background images. The statistical approach is used in order to exploit temporal as well as spatial image redundancies. The emphasis is directed towards the design of recursive and hybrid Bayesian filters.

In the recent past considerable attention has been devoted to the application of Kalman filtering to smoothing of observation noise in image data. Habibi [4] first suggested a two-dimensional recursive image estimator as an extension of the one-dimensional Kalman filter. Two other similar extensions were proposed [3], [5]. It has been shown [10] that these filters do not preserve the optimality of the one-dimensional Kalman filter. The most complete study of optimal two-dimensional Kalman filtering has been performed by Woods and Radewan [11]. They point out that the generalization of the Kalman filter to two dimensions can be done optimally with an extremely high dimensional state vector, which has dimensions on the order of MN (M = order of the filter, N = width of the image). Panda and Kak [9] succeeded in deriving a vector dynamic model that generates

the same random field of Habibi's model. Since this model is recursive in only one index, the one-dimensional Kalman filter is applied and, of course, optimality is preserved. Again, as indicated by Woods and Radewan [11], an extremely high dimensional state vector is used, which has the same dimension as the width of the image.

Several techniques have been investigated for background clutter suppression and target enhancement. Statistical non-recursive spatial filters [6] are suggested for background clutter suppression and enhancement of targets of known shapes. Two-dimensional Kalman filtering [5] is also proposed for the case of targets with arbitrary shapes. These filters assume that the image gray level can be decomposed into three additive components: target, background and white observation noise. The statistical differences of these three components are used in order to extract the target, which is the desired information.

B. RESEARCH OBJECTIVES

In the context of picture restoration and target detection as defined above, several specific research objectives have been identified.

a) Since the model of images plays a fundamental role in the statistical approach of image processing, several experiments with real life pictures will be accomplished in order to validate existing models [1-2] and to suggest new ones.

b) Generalization of the one-dimensional Kalman filter to two dimensions results in excessive computation loads. On the other hand, the filters [3-5] are very simple and, although it is known that they are not optimal, it was not determined yet just how far they are from optimality. Therefore, these filters will be compared against the optimum non-recursive filter.

c) The existing recursive image filters exploit only the spatial correlation, therefore a three-dimensional recursive filter will be developed in order to take advantage of the correlation in time.

d) The possibility of improving the performance of the sub-optimum recursive filters by using the non-causal (with respect to the direction of recursion) observations closest to the estimated pixel (picture element) will be investigated. The result is a hybrid filter in the sense of using some observations recursively and others non-recursively.

e) An optimum decision rule will be derived for purposes of background suppression and target enhancement and subsequent threshold detection.

f) A conventional Kalman filter will be constructed to track the target centroid from frame-to-frame.

C. OVERVIEW

The image modeling problem is discussed in Chapter II. First, a tutorial discussion of the mathematical problem of finding the dynamic model of random fields is presented.

Second, the results of several experiments with real life pictures are presented. Third, a model for a picture and for pictures sequenced in time is introduced.

In Chapter III the sub-optimum two-dimensional recursive filters [3-5] are analyzed. The exact computation of the error variance is particularly important to those filters, because they either do not compute the error variance in the calculation of the gains, or use an approximation. An exact method is developed to compute the error variance of those filters. Using such a method, the filters are compared among themselves and against the optimum non-recursive interpolator, constrained to the same data set. A recursive filter is also introduced that is essentially the same as [5], but the computation of gains is accomplished without approximation.

In Chapter IV a three-dimensional recursive filter is developed. This filter estimates the pixel gray level of pictures sequenced in time by using recursively the observations of the estimated frame, as well as those in the previous frames. It is an extension of the two-dimensional recursive filter [3]. Numerical results are presented to evaluate the improvement resulting from the exploitation of the correlation in time.

Chapter V introduces a new class of image filters, called hybrid filters. These filters are smoothers that

combine optimally the estimate of the recursive filters (two or three dimensions) with an arbitrary set of "future" observations. Theoretical comparison between the hybrid filter and the recursive [3] and non-recursive [6] filters is accomplished. Experiments with real life pictures are also presented for the purpose of comparison of these filters.

In Chapter VI an optimum decision rule is developed to detect targets immersed in cluttered background images. The target is considered as another texture statistically distinct from the background texture. The decision is made pixel by pixel and, therefore, it is independent of the target shape, but it can also be applied to targets with known shapes. The decision is based on the observation of the pixel gray level and the background prediction for the pixel. This prediction is given by the recursive or hybrid filters and may also be given by a non-recursive filter. Some special cases are worked out and its performance evaluated. The image is modeled as a weighted addition of three components: target, background and observation noise.

In Chapter VII a conventional Kalman filter is constructed to track the target centroid (or other points) from frame to frame. The target dynamics in the picture is modeled. The detector developed in Chapter VI feeds the tracking filter with the observations of the centroid spatial coordinates. The observation error is analyzed.

Chapter VIII presents several results of the simulation of a complete target detection and tracking problem, using computer generated images. Both recursive and hybrid filters are used and compared.

The final chapter summarizes the results of this investigation and presents the conclusions and suggestions for further study.

II. IMAGE MODELING

A. INTRODUCTION

Our objective in this chapter is to obtain a statistical model for pictures. Knowing part of a picture, one can generally draw certain inferences about the remainder; or, knowing a sequence of frames, one can, on the average, make a good guess or prediction about the next frame. From a statistical viewpoint, similarity between adjacent pixels (picture elements), or , frame-to-frame similarity represent a high level of intraframe or interframe correlation.

Experimental evicence [1]-[2] indicates that a monochromatic image can be modeled by specifying its value (gray level $x(m,n)$ at each spatial coordinate (m,n) . An ensemble of such images can be modeled by interpreting $x(m,n)$ as a random field.

In this chapter, first, we will address the mathematical problem of finding the dynamic model of random fields, given the autocorrelation function. Second, we will introduce a model for a picture and for pictures sequenced in time. Third, we will present experimental results in order to validate the proposed model.

B. DYNAMIC MODEL

Assume the mean and autocorrelation function of a homogeneous (wide sense stationary) random field are given. Since we are assuming knowledge of the mean, for convenience,

we assume the random field has zero mean, therefore autocorrelation and autocovariance are the same.

A highly desirable characteristic of any dynamic model is to have it excited by uncorrelated noise. The impulse response method always has such characteristics. The difficulty with this method is that the power spectrum density has to be factored, and there is no mathematical theorem for factorization in the multi-dimensional case. Since there exists considerable evidence, in refs. [1]-[2] and in this research, that images are well modeled by autocorrelation functions with separable kernels, the difficulty of factorization will not arise and, therefore, the impulse response method will be used.

The impulse response method is shown in figure 2.1. Assume there exists a system transfer function $H(z)$, such that, when driven by white noise $W(n)$, the resultant output $X(n)$ is a random process with the desired autocorrelation function $R(k)$. Using input-output relationships between power spectrum densities (PSD):

$$S_o(z) = S_i(z)H(z)H(z^{-1})$$

Since the input is white noise, its PSD is constant, $S_i(z) = K$, thus:

$$H(z)H(z^{-1}) = \frac{1}{K}S_o(z) = \frac{1}{K}Z(R(k))$$

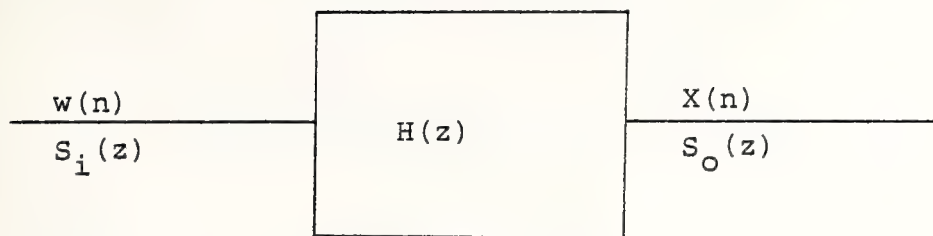


Figure 2.1

Filter Response Method

Therefore, to compute $H(z)$ we have to factor the right hand side of the previous equation. This is always possible, in the one dimensional case, provided $S_o(z)$ is a ratio of polynomials.

In the remainder of this section we present some examples to illustrate the method, as well as for later use.

Example 1

Consider the autocorrelation function:

$$R(k) = \sigma^2 e^{-\alpha |k|} \quad k = 0, \pm 1, \pm 2, \dots$$

let

$$\rho = e^{-\alpha}$$

$$R(k) = \sigma^2 \rho^{|k|} \quad (2.1)$$

Take the two-sided Z-transform of $R(k)$,

$$\begin{aligned}
Z(R(k)) &= \sum_{k=-\infty}^{\infty} R(k) z^{-k} \\
&= \sigma^2 (\dots + \rho^2 z^2 + \rho z + 1 + \rho z^{-1} + \rho^2 z^{-2} + \dots) \\
Z(R(k)) &= \frac{\sigma^2 (1-\rho^2)}{(1-\rho z^{-1})(1-\rho z)} \quad (2.2)
\end{aligned}$$

$$S_i(z)H(z)H(z^{-1}) = \frac{\sigma^2 (1-\rho^2)}{(1-\rho z^{-1})(1-\rho z)}$$

To have a stable model the poles of $H(z)$ must be inside the unit circle, therefore we choose:

$$H(z) = \frac{1}{1-\rho z^{-1}} = \frac{X(z)}{W(z)}$$

$$S_i(z) = \sigma^2 (1-\rho^2)$$

Thus, the dynamic model is:

$$X(n) = \rho X(n-1) + W(n) \quad (2.3)$$

Initializing equation (2.3) with $X(0)$, and exciting with white noise, having variance $\sigma^2 (1-\rho^2)$, a random process $X(n)$ is generated having the desired autocorrelation function.

Example 2

Consider the autocorrelation function:

$$R(k) = \frac{\sigma^2}{(\rho_1 - \rho_2)(1 + \rho_1 \rho_2)} [(1 - \rho_2^2) \rho_1^{|k|+1} - (1 - \rho_1^2) \rho_2^{|k|+1}] \quad (2.4)$$

where:

$$\rho_1 = e^{-\alpha_1}, \quad \rho_2 = e^{-\alpha_2}, \quad k = 0, \pm 1, \pm 2, \dots$$

Using results of example 1, the Z-transform of $R(k)$ is:

$$Z(R(k)) = \frac{Q}{(1 - \rho_1 z^{-1})(1 - \rho_2 z^{-1})(1 - \rho_1 z)(1 - \rho_2 z)} \quad (2.5)$$

where:

$$Q = \frac{\sigma^2(1 - \rho_1 \rho_2)}{1 + \rho_1 \rho_2} (1 - \rho_1^2)(1 - \rho_2^2)$$

Let's choose:

$$H(z) = \frac{X(z)}{W(z)} = \frac{1}{(1 - \rho_1 z^{-1})(1 - \rho_2 z^{-1})}$$

$$S_1(z) = Q$$

Thus, the dynamic model is:

$$X(n) = (\rho_1 + \rho_2)X(n-1) - \rho_1 \rho_2 X(n-2) + W(n) \quad (2.6)$$

It can be easily shown that a particular case of this model, when ρ_1 approaches ρ_2 , is:

$$R(k) = \sigma^2 \rho^{|k|} (1 + \beta |k|) \quad (2.7)$$

where

$$\beta = \frac{1 - \rho^2}{1 + \rho^2}, \quad \rho = e^{-\alpha}$$

and the dynamic model is:

$$X(n) = 2\rho X(n-1) - \rho^2 X(n-2) + W(N) \quad (2.8)$$

$$S_i(z) = \frac{\sigma^2 (1 - \rho^2)^3}{1 + \rho^2}$$

From equation (2.7) we can see that this autocorrelation function has zero derivative at the origin ($k = 0$) and an inflection point at $k = 1/\alpha$, provided that $\rho > 0.7$ or $\beta \approx \alpha$.

The model given by equation (2.4) is, therefore, a more general case and includes, as particular cases, the models given by equations (2.1) and (2.7). In figure 2.2 some curves of equation (2.4) are shown for the same "correlation time" (the point where the correlation is 37% of the maximum).

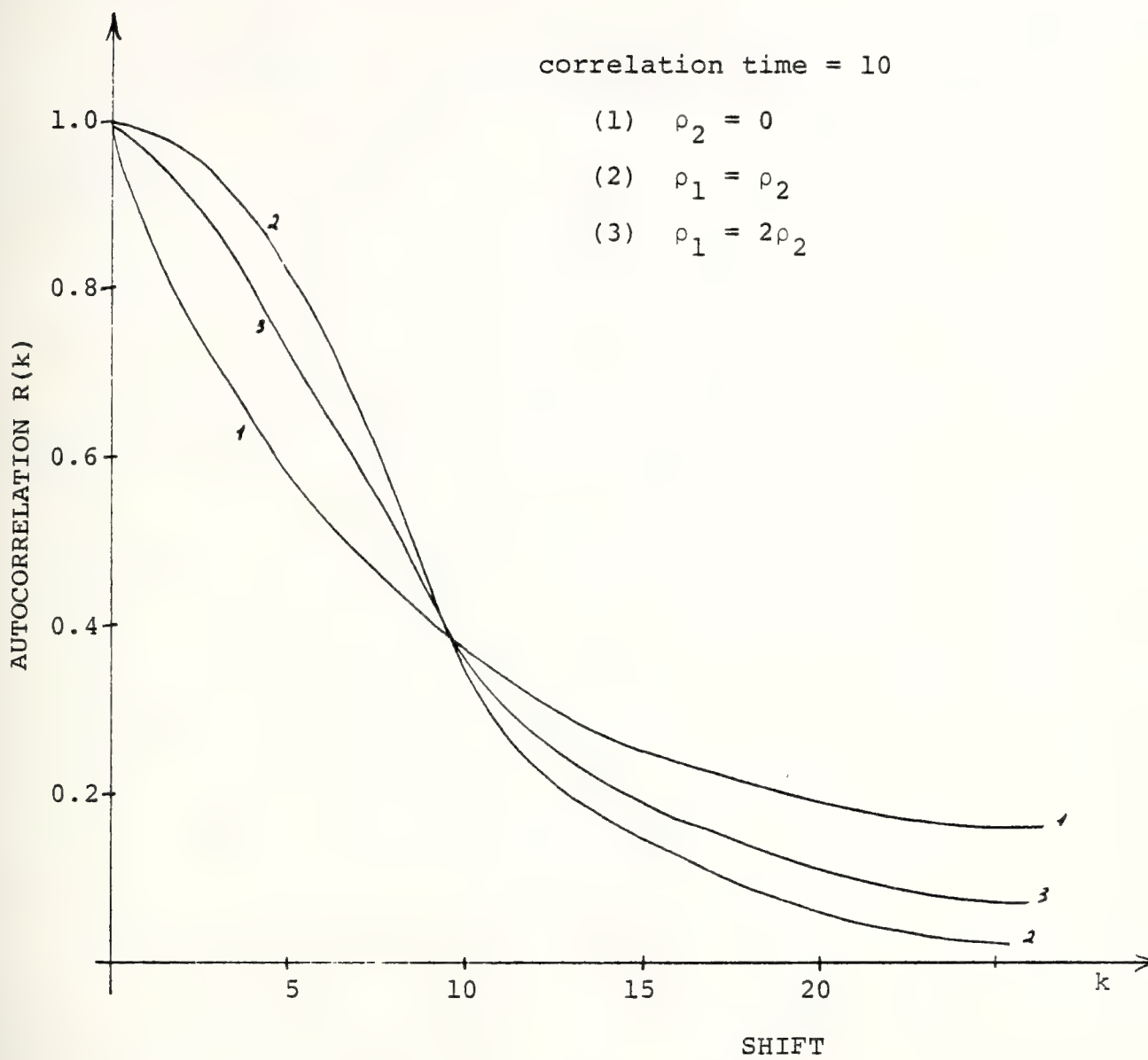


Figure 2.2
 Autocorrelation Functions

Example 3

Consider a two-dimensional autocorrelation function, separable in the two dimensions:

$$R(i,j) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \quad (2.9)$$

where

$$\rho_v = e^{-\alpha_v}, \quad \rho_h = e^{-\alpha_h}, \quad i,j = 0, \pm 1, \dots$$

To find the Z-transform we take advantage of separability and use equation (2.2) from example 1.

$$Z(R(i,j)) = \frac{\sigma^2 (1-\rho_v^2) (1-\rho_h^2)}{(1-\rho_v z_1^{-1}) (1-\rho_h z_2^{-1}) (1-\rho_v z_1) (1-\rho_h z_2)} \quad (2.10)$$

Let's choose:

$$H(z_1, z_2) = \frac{1}{(1-\rho_v z_1^{-1}) (1-\rho_h z_2^{-1})} = \frac{X(z_1, z_2)}{W(z_1, z_2)}$$

$$S_i(z_1, z_2) = \sigma^2 (1-\rho_v^2) (1-\rho_h^2)$$

Thus, the dynamic model is:

$$X(m,n) = \rho_v X(m-1,n) + \rho_h X(m,n-1) - \rho_v \rho_h X(m-1,n-1) + W(m,n) \quad (2.11)$$

Example 4

Similar to the previous example, consider a separable autocorrelation with the kernels of example 2:

$$R(i,j) = \sigma^2 K_v(i) K_h(j) \quad (2.12)$$

where:

$$K_v(i) = \frac{1}{(\rho_{1v} - \rho_{2v})(1 + \rho_{1v}\rho_{2v})} [(1 - \rho_{2v}^2)\rho_{1v}^{|i|+1} - (1 - \rho_{1v}^2)\rho_{2v}^{|i|+1}]$$

$$K_h(j) = \text{same equation with } i \rightarrow j \\ v \rightarrow h$$

The Z-transform is given by:

$$\begin{aligned} Z(R(i,j)) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} R(i,j) z_1^{-i} z_2^{-j} \\ &= \sigma^2 \sum_{i=-\infty}^{\infty} \left[\sum_{j=-\infty}^{\infty} K_h(j) z_2^{-j} \right] K_v(i) z_1^{-i} \\ &= \sigma^2 Z_1(K_v(i)) Z_2(K_h(j)) \end{aligned}$$

Using equation (2.5) it can be easily verified that the transfer function is:

$$H(z_1, z_2) = \frac{1}{(1-\rho_{1v}z_1^{-1})(1-\rho_{2v}z_1^{-1})(1-\rho_{1h}z_2^{-1})(1-\rho_{2h}z_2^{-1})} \quad (2.13)$$

$$S_i(z_1, z_2) = \sigma^2 Q_v Q_h$$

where:

$$Q_v = \frac{(1-\rho_{1v}\rho_{2v})}{1 + \rho_{1v}\rho_{2v}} (1-\rho_{1v}^2) (1-\rho_{2v}^2)$$

$$Q_h = \text{same equation with } v \rightarrow h$$

Thus, the dynamic model is:

$$\begin{aligned} X(m, n) = & (\rho_{1v} + \rho_{2v}) X(m-1, n) - \rho_{1v}\rho_{2v} X(m-2, n) \\ & + (\rho_{1h} + \rho_{2h}) X(m, n-1) - \rho_{1h}\rho_{2h} X(m, n-2) \\ & - (\rho_{1v} + \rho_{2v})(\rho_{1h} + \rho_{2h}) X(m-1, n-1) \\ & + \rho_{1v}\rho_{2v}(\rho_{1h} + \rho_{2h}) X(m-2, n-1) \\ & + \rho_{1h}\rho_{2h}(\rho_{1h} + \rho_{2v}) X(m-1, n-2) \\ & - \rho_{1v}\rho_{2v}\rho_{1h}\rho_{2h} X(m-2, n-2) + W(m, n) \end{aligned} \quad (2.14)$$

Observe that example 3 is a particular case of this, where $\rho_{2v} = \rho_{2h} = 0$.

To simplify notation, from now on, we will adopt the following convention:

$$X(m,n) = \underline{A} \underline{X} + W(m,n) \quad (2.15)$$

where:

\underline{A} = row vector of coefficients

\underline{X} = column vector of adjacents

In the case of example 3:

$$\underline{A} = [\rho_v \ \rho_h \ -\rho_v \rho_h]$$

$$\underline{X} = \begin{bmatrix} X(m-1,n) \\ X(m,n-1) \\ X(m-1,n-1) \end{bmatrix}$$

Example 5

Consider a three-dimensional separable autocorrelation function having kernels like examples 1 and 2:

$$R(i,j,k) = \sigma^2 K_1(k) K_2(i,j) \quad (2.16)$$

where:

$$K_1(k) = \rho_t^{|k|}$$

$$K_2(i,j) = K_v(i)K_h(j) \quad (\text{see example 4})$$

Following similar procedures as in example 4, it can be verified that the transfer function is:

$$H(z_1, z_2, z_3) = \frac{1}{(1-\rho_{1v}z_1^{-1})(1-\rho_{2v}z_1^{-1})(1-\rho_{1h}z_2^{-1})(1-\rho_{2h}z_2^{-1})(1-\rho_t z_3^{-1})} \quad (2.17)$$

$$S_i(z_1, z_2, z_3) = \sigma^2 Q_v Q_h (1-\rho_t^2)$$

where Q_v and Q_h are the same as in example 4.

Thus, the dynamic model is:

$$X(m,n,t) = \underline{A} \underline{X} + W(m,n,t) \quad (2.18)$$

where:

\underline{A} is a row vector whose elements are the coefficients of eq. 2.14, followed by these same coefficients multiplied by ρ_t , and the final element is ρ_t .

\underline{X} is a column vector whose elements are the adjacent pixels of eq. 2.14, followed by these same pixels located on the previous frame, and the final element is $X(m,n,t-1)$.

In figure 2.3 the adjacent pixels are shown.

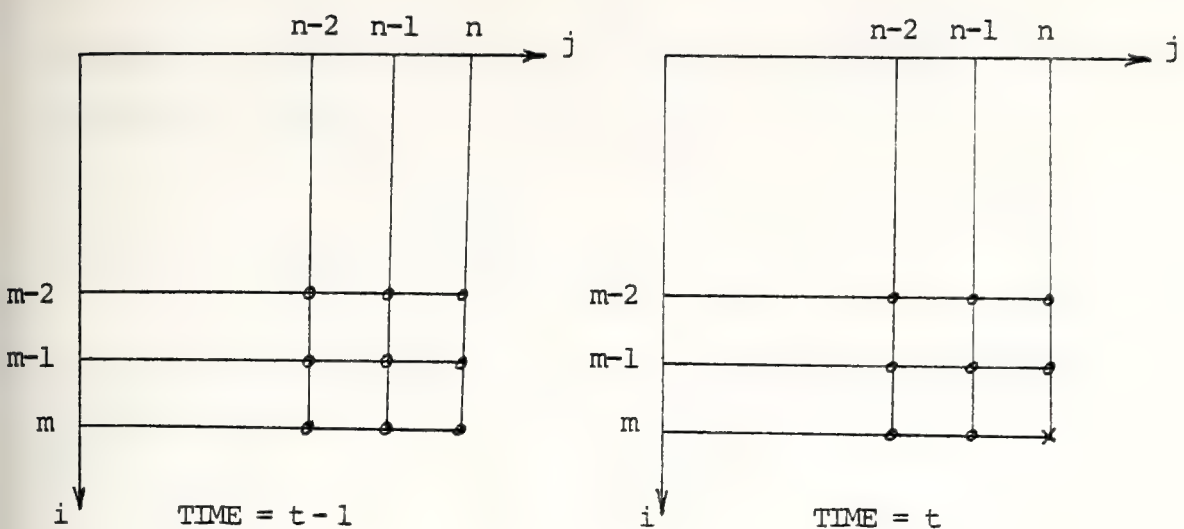


Figure 2.3

Adjacent pixels of example 5

C. AN AUTO-REGRESSIVE MODEL

In this section we present a stochastic model for images. The basic assumption is that images constitute a homogeneous random field with zero-mean (or known mean) and known autocorrelation function, separable in the independent dimensions. These assumptions will be validated by experiments in the next section.

1. Autocorrelation Function

The autocorrelation function chosen is intended to be general enough to include most of the models used in other research [2] - [9], as well as to best fit the experimental functions measured in this research and [1]-[2]. The hypothesis of separability allows us to examine the autocorrelation kernel by kernel. Assume that the process is

modeled by a kernel of first order, as in Example 1.

The dynamic model is

$$X(n) = \rho_1 X(n-1) + W_1(n)$$

If this model is valid, the modeling error $W_1(n)$ must be uncorrelated noise. This can be easily verified by computing the autocorrelation of $W_1(n)$ as follows. We compute $W_1(n)$ at each point by

$$W_1(n) = X(n) - \rho_1 X(n-1)$$

and then compute the autocorrelation of this sequence.

Computing the autocorrelation of $W_1(n)$ is a very good test. Assume that $W_1(n)$ turns out to be correlated.

In this case we can model $W_1(n)$ again by a first order kernel:

$$W_1(n) = \rho_2 W_1(n-1) + W_2(n)$$

Assume that, after repeating the same measurements for $W_2(n)$, we conclude that $W_2(n)$ is uncorrelated noise.

In figure 2.4 it is shown, in the Z-domain, the operations that we have performed.

Thus, the overall transfer function is:

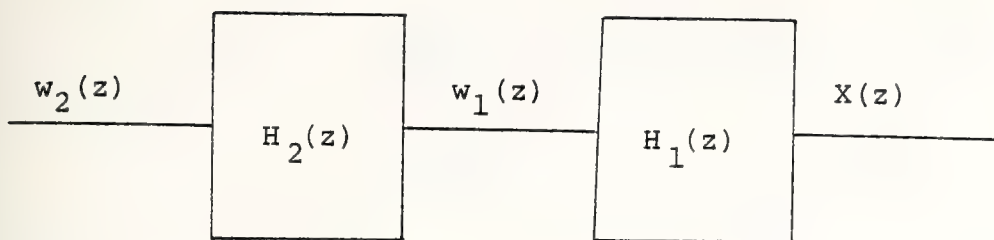


Figure 2.4
Second-order model

$$H(z) = \frac{X(z)}{W_2(z)} = \frac{1}{(1-\rho_1 z^{-1})(1-\rho_2 z^{-1})}$$

The second-order dynamic model is:

$$X(n) = (\rho_1 + \rho_2)X(n-1) - \rho_1 \rho_2 X(n-2) + W_2(n) \quad (2.19)$$

We have now a model driven by white noise, which is a second-order difference equation.

The process generated by this model has an autocorrelation function given by equation (2.4), as we will demonstrate.

$$E(W_2^2(n)) = Q_2$$

$$\begin{aligned}
Z(R(k)) &= Q_2 H(z) H(z^{-1}) \\
&= \frac{Q_2}{(1-\rho_1 z^{-1})(1-\rho_1 z)(1-\rho_2 z^{-1})(1-\rho_2 z)} \\
&= \frac{AQ_2}{(1-\rho_1 z^{-1})(1-\rho_1 z)} + \frac{BQ_2}{(1-\rho_2 z^{-1})(1-\rho_2 z)}
\end{aligned}$$

where:

$$\begin{aligned}
A &= \frac{\rho_1}{(\rho_1 - \rho_2)(1 - \rho_1 \rho_2)} \\
B &= \frac{-\rho_2}{(\rho_1 - \rho_2)(1 - \rho_1 \rho_2)}
\end{aligned}$$

since

$$\begin{aligned}
z^{-1} \left(\frac{1}{(1-\rho z^{-1})(1-\rho z)} \right) &= \frac{1}{1-\rho^2} \rho^{|k|} \\
R(k) &= Q_2 \left[\frac{A}{1-\rho_1^2} \rho_1^{|k|} - \frac{B}{1-\rho_2^2} \rho_2^{|k|} \right]
\end{aligned}$$

Using $R(0) = \sigma^2$ (variance of $X(n)$)

$$Q_2 = \sigma^2 (1-\rho_1^2)(1-\rho_2^2) \frac{(1-\rho_1 \rho_2)}{(1+\rho_1 \rho_2)} \quad (2.20)$$

$$\begin{aligned}
R(k) &= \frac{\sigma^2}{(\rho_1 - \rho_2)(1 + \rho_1 \rho_2)} [\rho_1 (1-\rho_2^2) \rho_1^{|k|} - \rho_2 (1-\rho_1^2) \rho_2^{|k|}] \\
&\quad (2.21)
\end{aligned}$$

Assume, now, that $W_2(n)$ is not uncorrelated enough. In such a case we proceed to a third-order model. Following similar procedures as for the second-order model, we have:

Dynamic Model:

$$\begin{aligned} X(n) = & (\rho_1 + \rho_2 + \rho_3)X(n-1) - (\rho_1\rho_2 + \rho_1\rho_3 + \rho_2\rho_3)X(n-2) \\ & + \rho_1\rho_2\rho_3X(n-3) + W_3(n) \end{aligned} \quad (2.22)$$

Autocorrelation function:

$$R(k) = \sigma^2 \frac{A_1 \rho_1^{|k|} + A_2 \rho_2^{|k|} + A_3 \rho_3^{|k|}}{A_1 + A_2 + A_3} \quad (2.23)$$

where:

$$A_1 = \rho_1^2 (\rho_2 - \rho_3) (-\rho_2 \rho_3) (1 - \rho_2^2) (1 - \rho_3^2)$$

$$A_2 = -\rho_2^2 (\rho_1 - \rho_3) (1 - \rho_1 \rho_3) (1 - \rho_1^2) (1 - \rho_3^2)$$

$$A_3 = \rho_3^2 (\rho_1 - \rho_2) (1 - \rho_1 \rho_2) (1 - \rho_1^2) (1 - \rho_2^2)$$

Variance of $W_3(n)$:

$$Q_3 = \frac{\sigma^2}{\frac{A_1}{1 - \rho_1^2} + \frac{A_2}{1 - \rho_2^2} + \frac{A_3}{1 - \rho_3^2}} \quad (2.24)$$

The general expressions for a k-order model are:

Dynamic model:

$$\begin{aligned}
 X(n) = & (\rho_1 + \rho_2 + \dots + \rho_k)X(n-1) - (\rho_1\rho_2 + \rho_1\rho_3 + \dots + \rho_{k-1}\rho_k)X(n-2) \\
 & + (\rho_1\rho_2\rho_3 + \dots + \rho_{k-2}\rho_{k-1}\rho_k)X(n-3) \\
 & + (-1)^{k+1}\rho_1\rho_2\dots\rho_kX(n-k) + W_k(n)
 \end{aligned} \tag{2.25}$$

Autocorrelation function:

$$R(i) = \sigma^2 \frac{A_1\rho_1^{|i|} + A_2\rho_2^{|i|} + \dots + A_k\rho_k^{|i|}}{A_1 + A_2 + \dots + A_k} \tag{2.26}$$

Variance of $W_k(n)$:

$$Q_k = \frac{\sigma^2}{\frac{A_1}{1-\rho_1^2} + \frac{A_2}{1-\rho_2^2} + \dots + \frac{A_k}{1-\rho_k^2}} \tag{2.27}$$

The expressions for the A's are easily inferred from equation (2.23).

In example 3 an auto-regressive model is presented for a picture with an autocorrelation whose kernels in the two spatial dimensions are the first order versions ($k=1$) of equation (2.26). In fact, that model is used in most research [2] - [9] with images. In example 4 an image model is presented whose kernels are the second-order versions ($k=2$) of equation (2.26). The order of the

kernels doesn't have to be the same. Depending on the particular picture to be modeled, the order is chosen for each direction.

In example 5 an auto-regressive model is presented for pictures sequenced in time. In that case we chose second-order kernels in the spatial directions and first-order in the time direction.

At this point, some special features of the proposed autocorrelation functions given by equation (2.26) should be emphasized:

a) It includes as a particular case the first order model which is used in most research [2] - [9].

b) It fits much better many situations where the first order model is a poor approximation.

c) The resultant dynamic model is auto-regressive and driven by white noise.

d) The cascade feature of whitening the modeling error is quite simple and adequate for on line parameter identification. This feature will be exploited in the next item.

2. Parameter Identification

Given that the autocorrelation function of an image, or a time-frame of images, is assumed to have separable kernels as in equation (2.26), the next problem is to find its parameters. Let's discuss the modeling of just a picture, since the extension for the time-frame is

straightforward. Having a specific picture, we want to identify the parameters of the model. To compute statistics from just one realization of the process we must assume the process is ergodic, therefore we consider the picture as a typical realization and the averaging in space is equivalent to statistical averaging. Also the size of the picture has to be sufficiently large.

Those conditions are quite severe, and we realize that the process has to be restricted to a particular class of pictures with common properties. Another way of looking at this is to visualize the picture as an array, for example, of 100×100 , therefore with 10,000 pixels. Each pixel can have, say, 100 distinguishable brightness values. The number of different pictures is $100^{10,000}$. With such enormous numbers of pictures, even if we were able to compute the mean and autocorrelation, such moments would not be enough information. It would be necessary to know statistics of enormously higher orders. The attention, therefore, has to be focused on local redundancy, encompassing only a few pictures closely related. To have a model as accurate as possible, based only in the first and second moments, we will need some degree of adaptation.

Our method is based on the cascade feature of equation (2.26). Let's explain the method in detail. Assume a picture is given with negligible measurement noise. We proceed in the following steps.

Step 1 - Compute the mean and remove it from the picture.

Step 2 - Compute the autocorrelation in the horizontal direction, with a number of displacements equal to 20% of the number of columns. There is no use to go much further than this, because the accuracy in the averaging is reduced. Repeat the same procedure in the vertical direction. This step can be accomplished using the Fast Fourier Transform algorithm (FFT).

Step 3 - Having the measured autocorrelation in the horizontal and vertical directions, assume that equation (2.26) is adequate and gives a nice fit. Start with a first-order kernel and compute ρ_{1v} and ρ_{1h} , such that the mean-squared error in the fitting is minimized. The result is a model like example 3:

$$X(m,n) = \underline{A}_1 \underline{X} + W_1(m,n)$$

where:

$$\underline{A}_1 = [\rho_{1v} \quad \rho_{1h} \quad -\rho_{1v}\rho_{1h}]$$

$$\underline{X}^T = [X(m-1,n) \quad X(m,n-1) \quad X(m-1,n-1)]$$

Step 4 - At this point we have a model for the picture that is first-order in both spatial directions. To test the goodness of this model the obvious figure of merit is the modeling error $W_1(m,n)$. If the model is adequate, $W_1(m,n)$

should be uncorrelated noise. To test this, compute the autocorrelation of $W_1(m,n)$ as in step 2. If it is correlated in both directions, it means the first-order model is not good enough and we have to proceed to a second-order model. It could happen that we are satisfied with one direction, but not with the other, in this case we have to proceed to a second-order only in such directions. Assume here that $W_1(m,n)$ is correlated in both directions.

Step 5 - Repeat step 3 for $W_1(m,n)$ in order to find ρ_{2v} and ρ_{2h} . The model for $W_1(m,n)$ is:

$$W_1(m,n) = \underline{A}_2 \underline{W}_1 + W_2(m,n)$$

where:

$$\underline{A}_2 = [\rho_{2v} \quad \rho_{2h} \quad -\rho_{2v}\rho_{2h}]$$

$$\underline{W}_1^T = [W_1(m-1,n) \quad W_1(m,n-1) \quad W_1(m-1,n-1)]$$

The second-order model for the picture is given by equation (2.14) in example 4.

Step 6 - Repeat step 4 for $W_2(m,n)$. If it is sufficiently uncorrelated, the procedure stops and we have identified the order of the model as well as the parameters. Otherwise we proceed seeking a higher order model.

NOTE: Step 4 is responsible for the decision to proceed or not and also in which directions, whether horizontal, vertical or both. The decision to proceed or not can be implemented only based on the variance of $W(m,n)$, by observing its reduction after the second-order modeling. If such reduction is null or negligible, it means the original model is good and it is not necessary to increase the order. The decision about the direction to be improved can be implemented by observing the autocorrelation of $W(m,n)$ at a few points, say, the first 5 displacements.

The computation of the coefficients is quite simple because we have reduced the problem to fitting exponentials at each step. Such computation is accomplished as follows. Assume the measured autocorrelation in one direction is given by $Y(n)$, $n = 0, 1, \dots, K$ and it is normalized such that $Y(0) = 1$. We want to fit an exponential minimizing the mean-squared error, $R(n) = e^{-\alpha n} = \rho^n$. Taking the natural logarithm of $R(n)$ and $Y(n)$, we want to minimize:

$$\epsilon^2 = \sum_{n=1}^K [\ln R(n) - \ln Y(n)]^2$$

$$\epsilon^2 = \sum_{n=1}^K [-\alpha n - \ln Y(n)]^2$$

Take the derivative with respect to α and equate to zero, the result being:

$$\alpha = \frac{\frac{\sum_{n=1}^K n \ln Y(n)}{K}}{\sum_{n=1}^K n^2} = \frac{-6 \sum_{n=1}^K n \ln Y(n)}{K(K+1)(2K+1)}$$

and

$$\rho = e^{-\alpha}.$$

In figure 2.5 a flowchart to implement the proposed method of modeling is shown.

To model pictures sequenced in time the procedure is quite similar. In this case we have a sequence of pictures, sequenced in time, and apply the method adding the extra dimension (time). Example 5 presents a model that is first-order in time and second-order in both spatial dimensions.

3. Modeling Of Noisy Images

In the last item, a method for picture modeling was presented in which the picture was assumed to be noise free. Such an assumption restricts considerably real life applications where there is no knowledge of the original picture. This restriction can be removed if the autocorrelation function of the measurement noise is known.

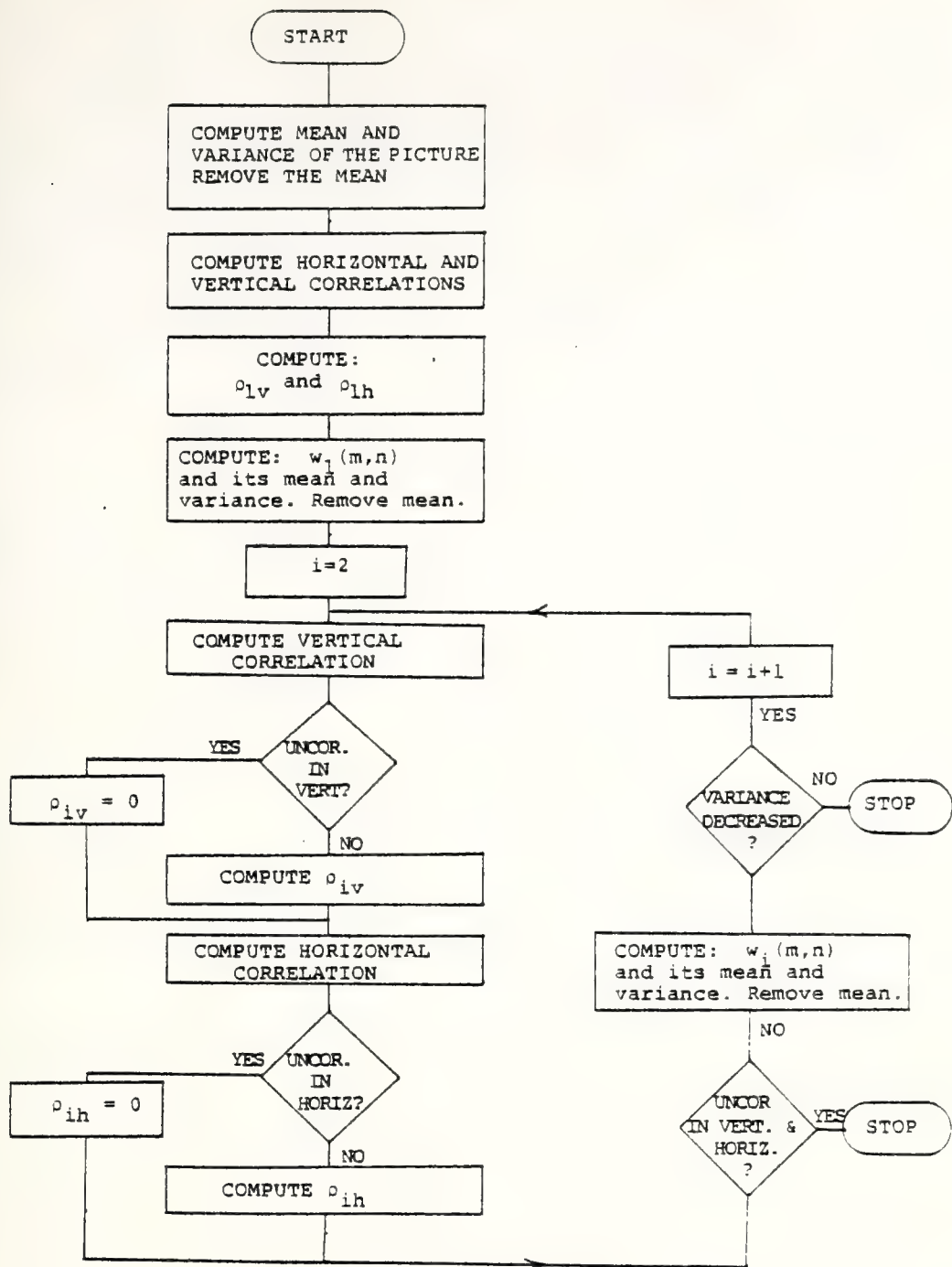


Figure 2.5

Parameter Identification Flowchart

In what follows, we present the steps to be modified in the previous modeling method, in order to take into account additive zero-mean white measurement noise, with known variance.

The measurement equation is:

$$y(m,n) = X(m,n) + v(m,n)$$

where $v(m,n)$ is zero-mean white noise, uncorrelated with $X(m,n)$ and with known variance:

$$E(v^2(m,n)) = r$$

Step 2 - Compute the autocorrelation of the noisy picture $y(m,n)$ in both directions, as before, but remove the variance R from $R_{YY}(0,0)$ to obtain $R_{XX}(0,0)$, since:

$$R_{XX}(i,j) = \begin{cases} R_{YY}(0,0) - r & (i,j) = (0,0) \\ R_{YY}(i,j) & (i,j) \neq (0,0) \end{cases}$$

In figure 2.6 the relation between the autocorrelations of $X(m,n)$ and $y(m,n)$ is illustrated in one direction.

Step 4 - Since $X(m,n)$ is not known, $W_1(m,n)$ can't be calculated, but its autocorrelation can be computed indirectly.

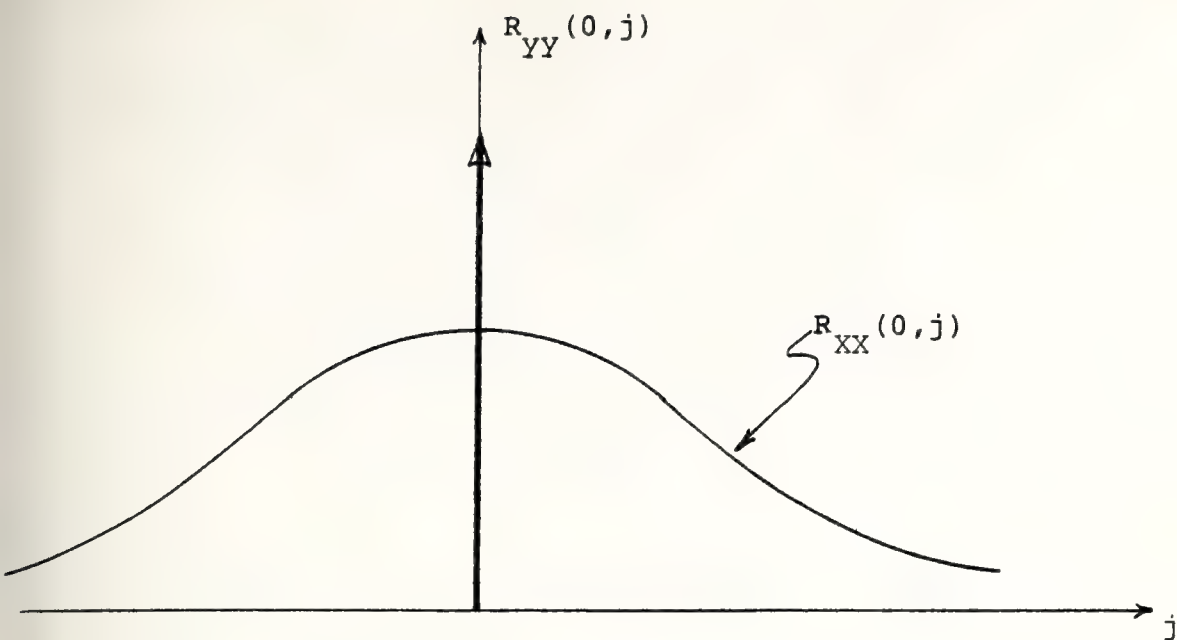


Figure 2.6
Autocorrelation of $Y(m,n)$

$$X(m,n) = \underline{A}_1 \underline{X} + W_1(m,n) \quad (2.28)$$

$$Y(m,n) = X(m,n) + v(m,n) \quad (2.29)$$

Substituting (2.28) into (2.29):

$$Y(m,n) = \underline{A}_1 \underline{Y} + v(m,n) + W_1(m,n) - \underline{A}_1 \underline{V}$$

where:

$$\underline{Y}^T = [Y(m-1,n) \quad Y(m,n-1) \quad Y(m-1,n-1)]$$

$$\underline{V}^T = [v(m-1,n) \quad v(m,n-1) \quad v(m-1,n-1)]$$

Let's call:

$$v_1(m,n) = v(m,n) - \underline{A}_1 \underline{V} \quad (2.30)$$

$$u_1(m,n) = W_1(m,n) + v_1(m,n)$$

$$y(m,n) = \underline{A}_1 \underline{Y} + u_1(m,n) \quad (2.31)$$

Since $y(m,n)$ and \underline{A}_1 are known, $u_1(m,n)$ can be computed, and, therefore, its autocorrelation in both directions.

Let's examine the autocorrelation of $u_1(m,n)$ in the horizontal direction.

$$E[u_1(m,n)u_1(m,n+j)] = E[v_1(m,n)v_1(m,n+j)] + E[W_1(m,n)W_1(m,n+j)]$$

$$E[v_1(m,n)v_1(m,n+j)] = E[v(m,n)v(m,n+j)] - \underline{A}_1 E[v(m,n)\underline{V}(m,n+j)]$$

$$- \underline{A}_1 E[v(m,n+j)\underline{V}(m,n)]$$

$$+ \underline{A}_1 E[\underline{V}(m,n)\underline{V}^T(m,n+j)]\underline{A}_1^T$$

where:

$$E[v(m,n)v(m,n+j)] = r\delta(0,j)$$

$$E[v(m,n)\underline{v}(m,n+j)] = \begin{bmatrix} 0 \\ r\delta(0,j-1) \\ 0 \end{bmatrix}$$

$$E[v(m,n+j)\underline{v}(m,n)] = \begin{bmatrix} 0 \\ r\delta(0,j+1) \\ 0 \end{bmatrix}$$

$$E[\underline{v}(m,n)\underline{v}^T(m,n+j)] = \begin{bmatrix} r\delta(0,j) & 0 & r\delta(0,j-1) \\ 0 & r\delta(0,j) & 0 \\ r\delta(0,j+1) & 0 & r\delta(0,j) \end{bmatrix}$$

Therefore:

$$\begin{aligned} R_{v_1 v_1}(0,j) &= r(1+\rho_{1v}^2 + \rho_{1h}^2 + \rho_{1v}^2 \rho_{1h}^2) \delta(0,j) \\ &\quad - r\rho_{1h}(1+\rho_{1v}^2) \delta(0,j-1) - r\rho_{1h}(1+\rho_{1v}^2) \delta(0,j+1) \end{aligned}$$

$$R_{u_1 u_1}(0,j) = R_{v_1 v_1}(0,j) + R_{w_1 w_1}(0,j)$$

or:

$$R_{w_1 w_1}(0,j) = \begin{cases} R_{u_1 u_1}(0,0) - r(1+\rho_{1v}^2 + \rho_{1h}^2 + \rho_{1v}^2 \rho_{1h}^2) \dots j = 0 \\ R_{u_1 u_1}(0,1) - r\rho_{1h}(1+\rho_{1v}^2) \dots |j| = 1 \\ R_{u_1 u_1}(0,j) \dots \text{elsewhere} \end{cases}$$

In figure 2.7 the relation between the autocorrelations of $u_1(m,n)$ and $w_1(m,n)$ is illustrated.

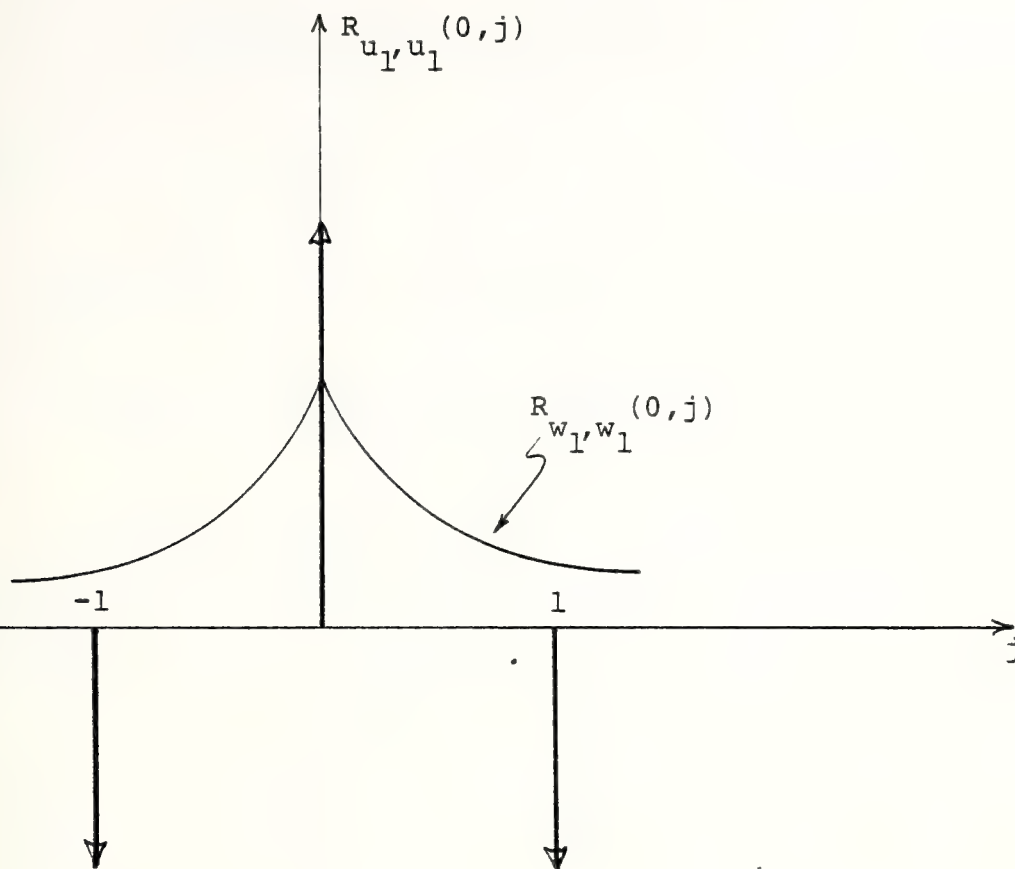


Figure 2.7

Autocorrelation of $u_1(m,n)$

The procedure above is easily extended to calculate the autocorrelation of $w_2(m,n)$, as well as higher order modeling errors.

$$w_1(m,n) = \underline{A}_2 \underline{w}_1 + w_2(m,n) \quad (2.32)$$

$$u_1(m,n) = W_1(m,n) + v_1(m,n) \quad (2.33)$$

Substitute (2.32) into (2.33):

$$u_1(m,n) = \underline{A}_2 \underline{U}_1 + u_2(m,n) \quad (2.34)$$

where:

$$u_2(m,n) = W_2(m,n) + v_2(m,n) \quad (2.35)$$

$$v_2(m,n) = v_1(m,n) - \underline{A}_2 \underline{V}_1$$

From equation (2.34) we can calculate $u_2(m,n)$, since $u_1(m,n)$ was computed by equation (2.31), and the coefficients \underline{A}_2 are given by the fitting of $R_{W_1 W_1}(i,j)$.

To continue the modeling, the autocorrelation of $W_2(m,n)$ is derived from $R_{u_2 u_2}(i,j)$ using equation (2.35).

D. EXPERIMENTAL RESULTS

In this section we present some relevant results obtained with real life pictures. Several pictures were analyzed and divided in two broad classes:

Picture A - represents a class of pictures with few details (i.e. a low content of high spatial frequency structure). In figure 2.8 a sample of such pictures in a three-dimensional plot is shown, where the height (z-axis) is the gray level. This kind of plot is very useful to emphasize small details.

Picture B - represents a class of pictures with many details. Figure 2.14 shows a sample of such pictures.

1. Autocorrelation Function

The proposed model assumes that pictures have separable autocorrelation functions whose kernels are given by equation (2.26). To validate this assumption, the autocorrelation of pictures A and B were measured and compared with that of the proposed model. Figure 2.9 is the measured autocorrelation of picture A, and figure 2.15 is the same for picture B. Using minimum mean-squared error criteria, a first-order model was fitted for both pictures, as shown in figures 2.10 and 2.16. Such a model didn't work well for picture A, as can be seen, comparing figures 2.9 and 2.10. For picture B it worked quite well, as can be seen comparing figures 2.15 and 2.16. Figure 2.11 is the second-order model for picture A; observe that it is a very good fit. Based on these results, we conclude:

a) The proposed autocorrelation is adequate for picture modeling.

b) The first-order model used in other research does not fit well pictures of class A.

c) Pictures with few details are best fitted by a second-order model.

d) Pictures with many details are adequately fitted by first-order model.

2. Parameter Identification

The proposed method of parameter identification was applied to both pictures. Figures 2.12 and 2.17 are the autocorrelation of $W_1(m,n)$ for pictures A and B, respectively. The autocorrelation of the modeling error $W_1(m,n)$ is the measurement of the "goodness" of the first-order model. As predicted before, picture A has a poor first-order model, since $W_1(m,n)$ is quite correlated, instead of white noise, as can be seen in figure 2.12. On the other hand, picture B has a nice first-order model, since $W_1(m,n)$ is almost white noise, as can be seen in figure 2.17.

Applying the cascade method of modeling for picture A, the second-order modeling error $W_2(m,n)$ was calculated and its autocorrelation plotted in figure 2.13. Now the modeling error $W_2(m,n)$ is quite close to white noise, and the second-order model is adequate for picture A, as predicted before.

3. Bandwidth Compression

In order to present another piece of evidence to validate the model, it was applied to bandwidth compression. Picture B is quantized in 256 gray levels, thus, 8 bits are required to transmit the gray level of each pixel. Exploiting redundancies in the picture and/or in a time frame of pictures (like television), it seems possible to reduce the number of bits/pixels to be transmitted, therefore, reducing the required bandwidth.

The model for picture B is:

$$X(m,n) = \underline{A} \underline{X} + W_1(m,n)$$

The measured variances of $X(m,n)$ and $W_1(m,n)$ are:

$$\text{VAR}(X) \doteq 49$$

$$\text{VAR}(W_1) \doteq 1$$

Also, it was seen that $W_1(m,n)$ is almost uncorrelated noise. Therefore, instead of transmitting the picture X , we can transmit W_1 :

$$W_1(m,n) = X(m,n) - \underline{A} \underline{X}$$

Since W_1 is close to white noise, and has much smaller variance than X , the quantization levels for W may be much smaller than for X . This method of bandwidth reduction is called DPCM (Differential Pulse-code Modulation).

Of course, the effectiveness of this method is strongly dependent on the "goodness" of the model.

Figure 2.18 presents the histogram of the modeling error W_1 . It can be seen that most "energy" of W_1 is concentrated around zero. In figure 2.19 the reconstruction of picture B is presented, where W_1 was transmitted with only 2 bits/pixel (average), instead of the 8 bits/pixel

required for X. Therefore, even with a bandwidth reduction of 4 times, the reconstructed picture yet carries most of the information of the original.

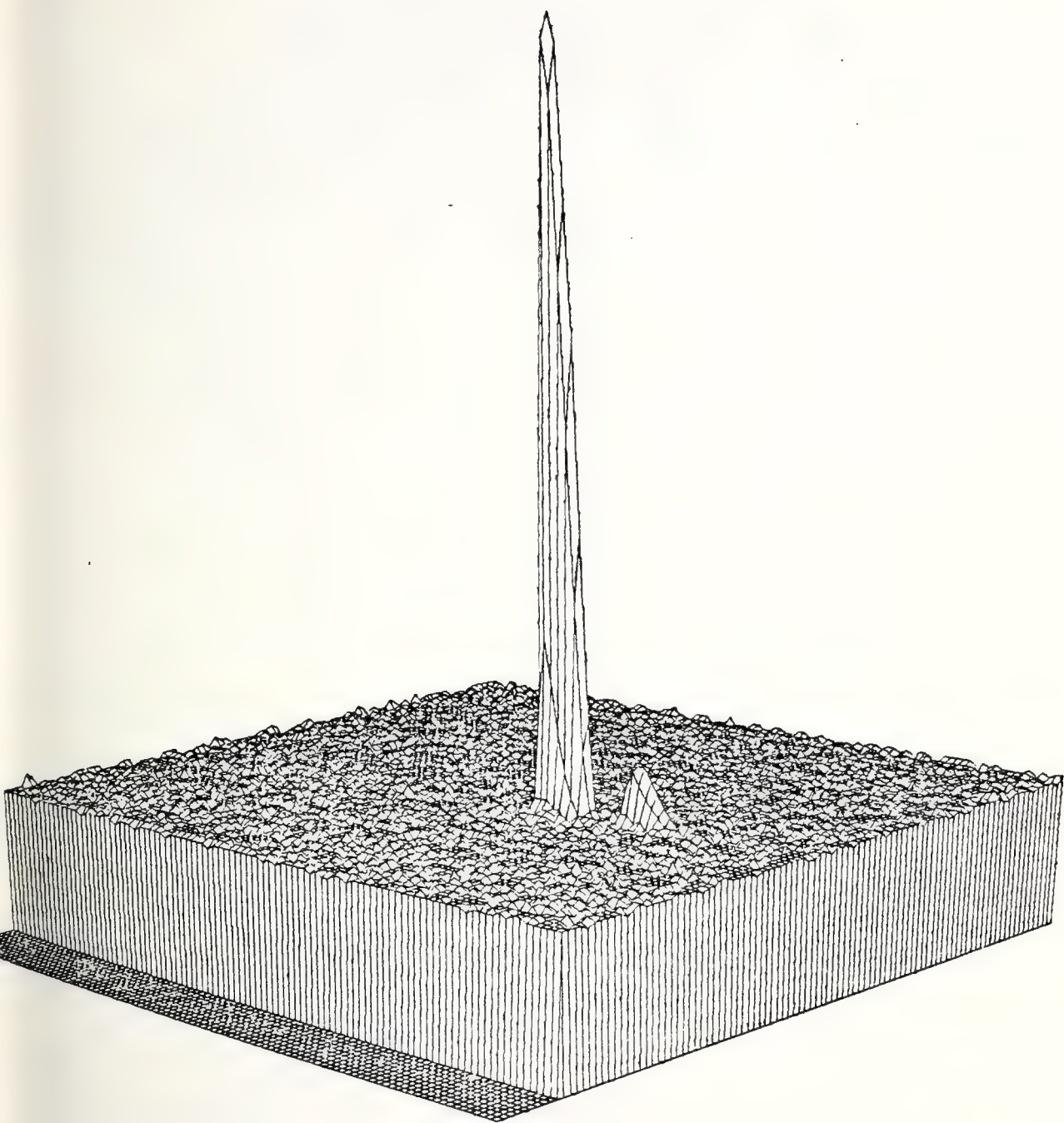


Figure 2.8

Picture A

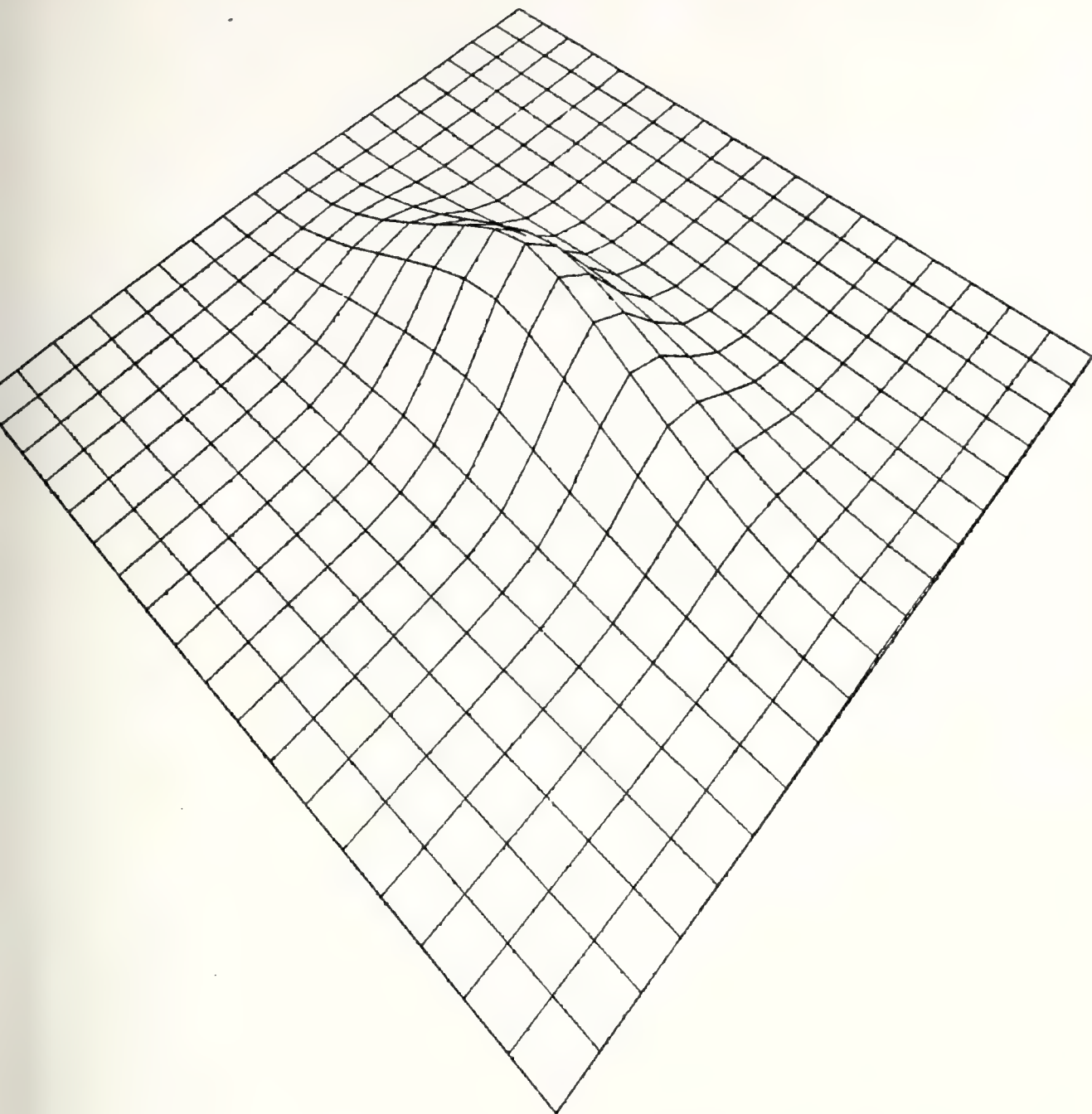


Figure 2.9

Measured Autocorrelation of Picture A

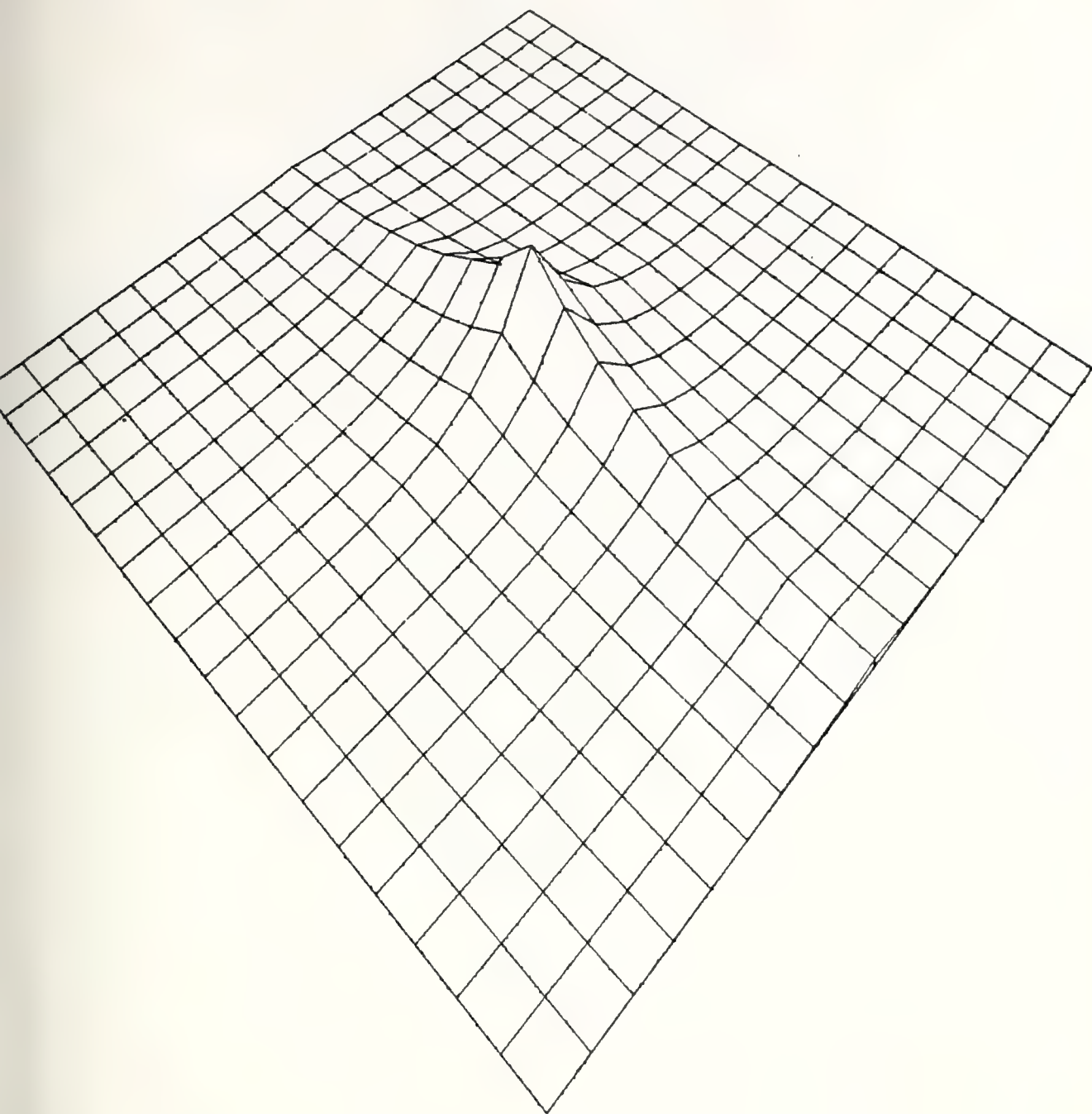


Figure 2.10
First Order Model of Picture A

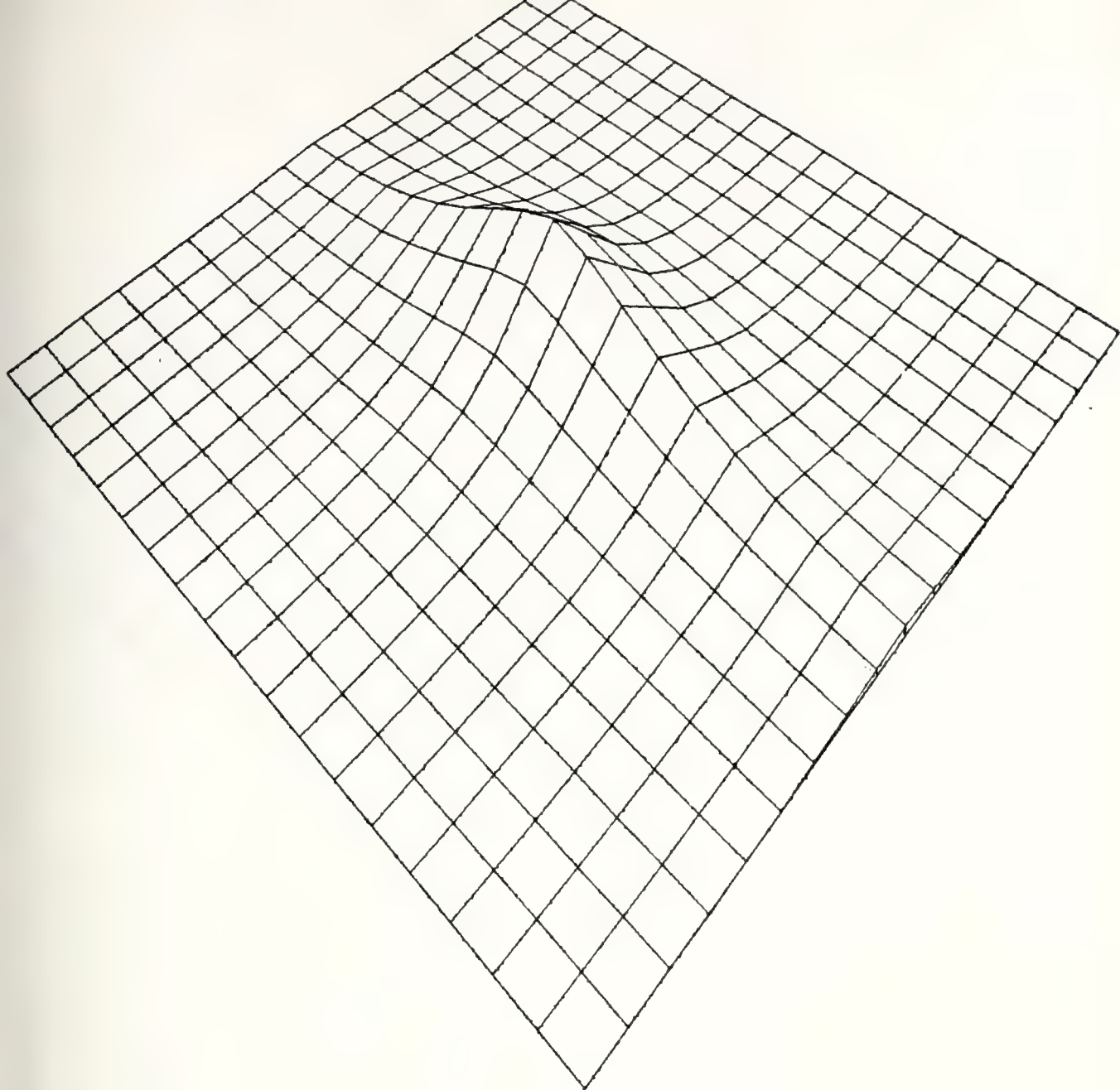


Figure 2.11
Second Order Model of Picture A

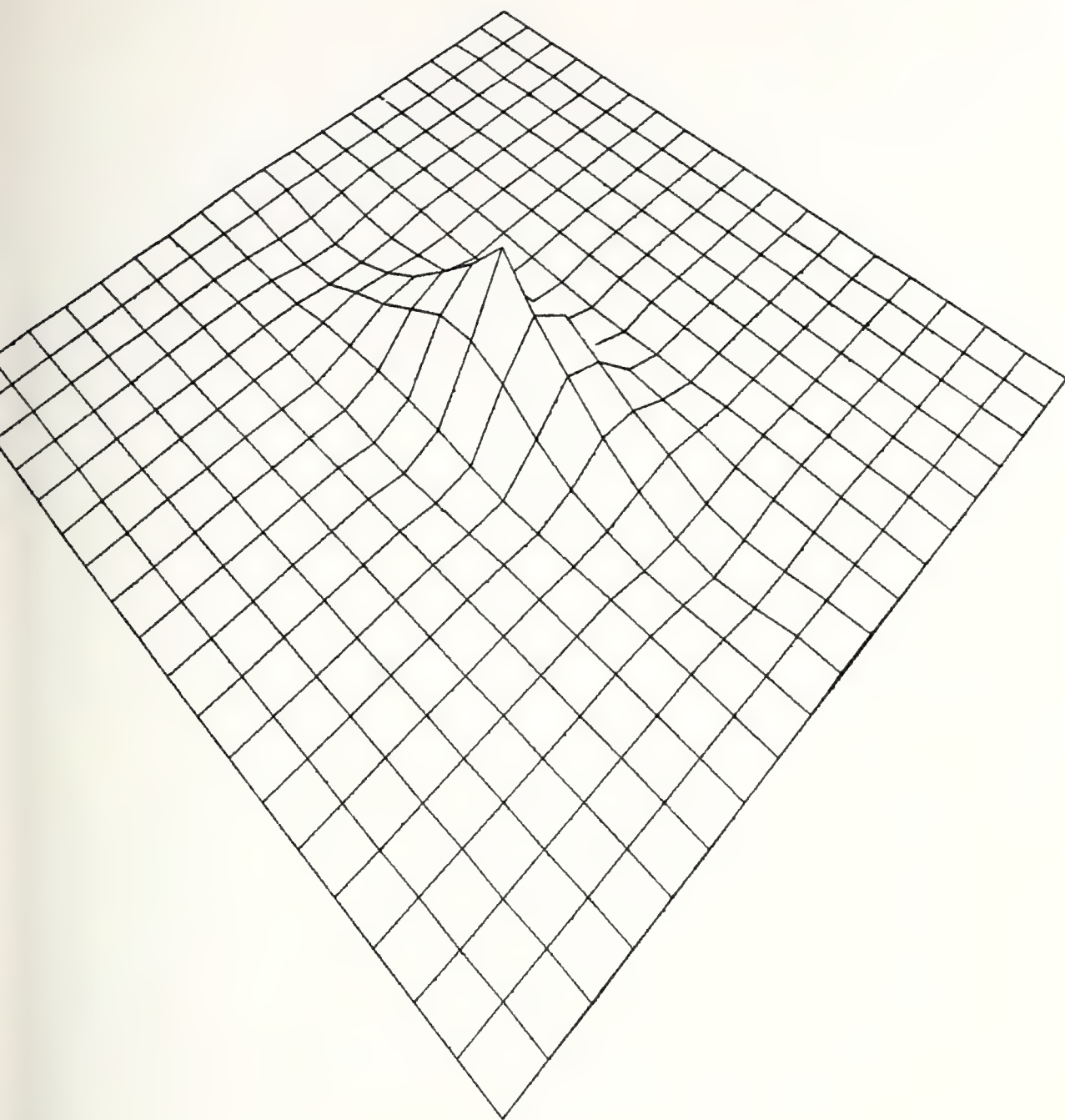


Figure 2.12
First Order Modeling Error of Picture A

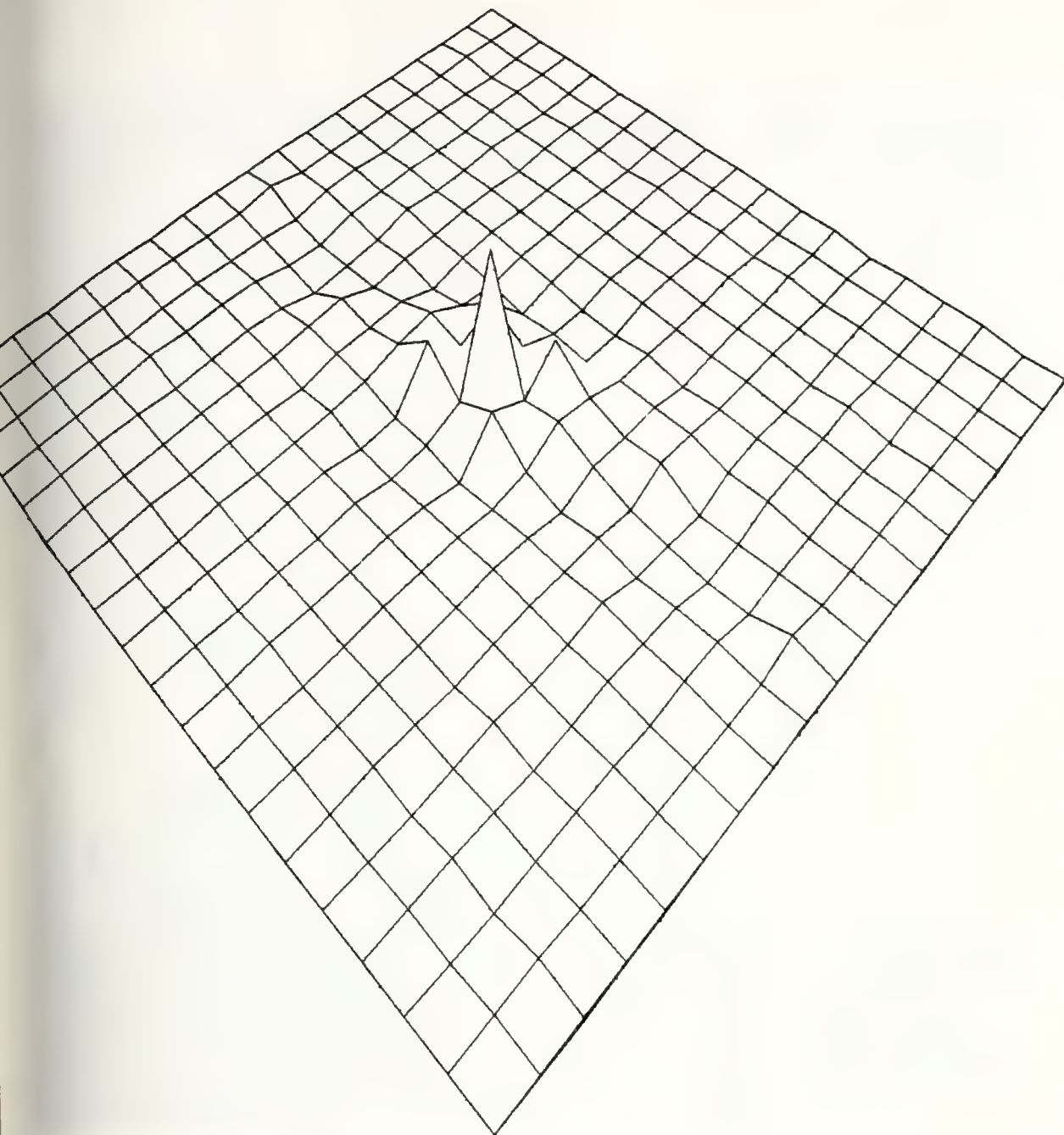


Figure 2.13
Second Order Modeling Error of Picture A

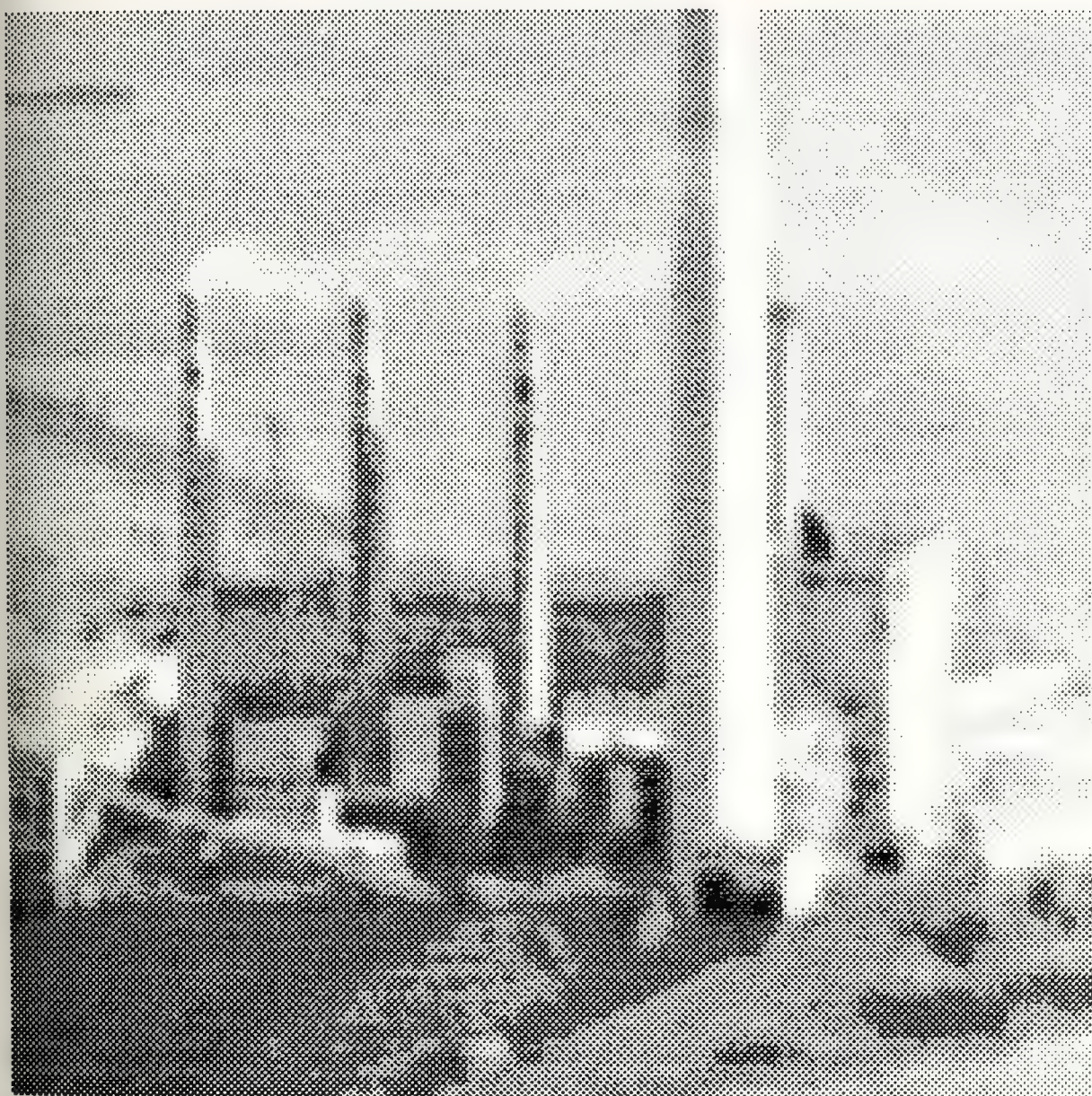


Figure 2.14

Picture B

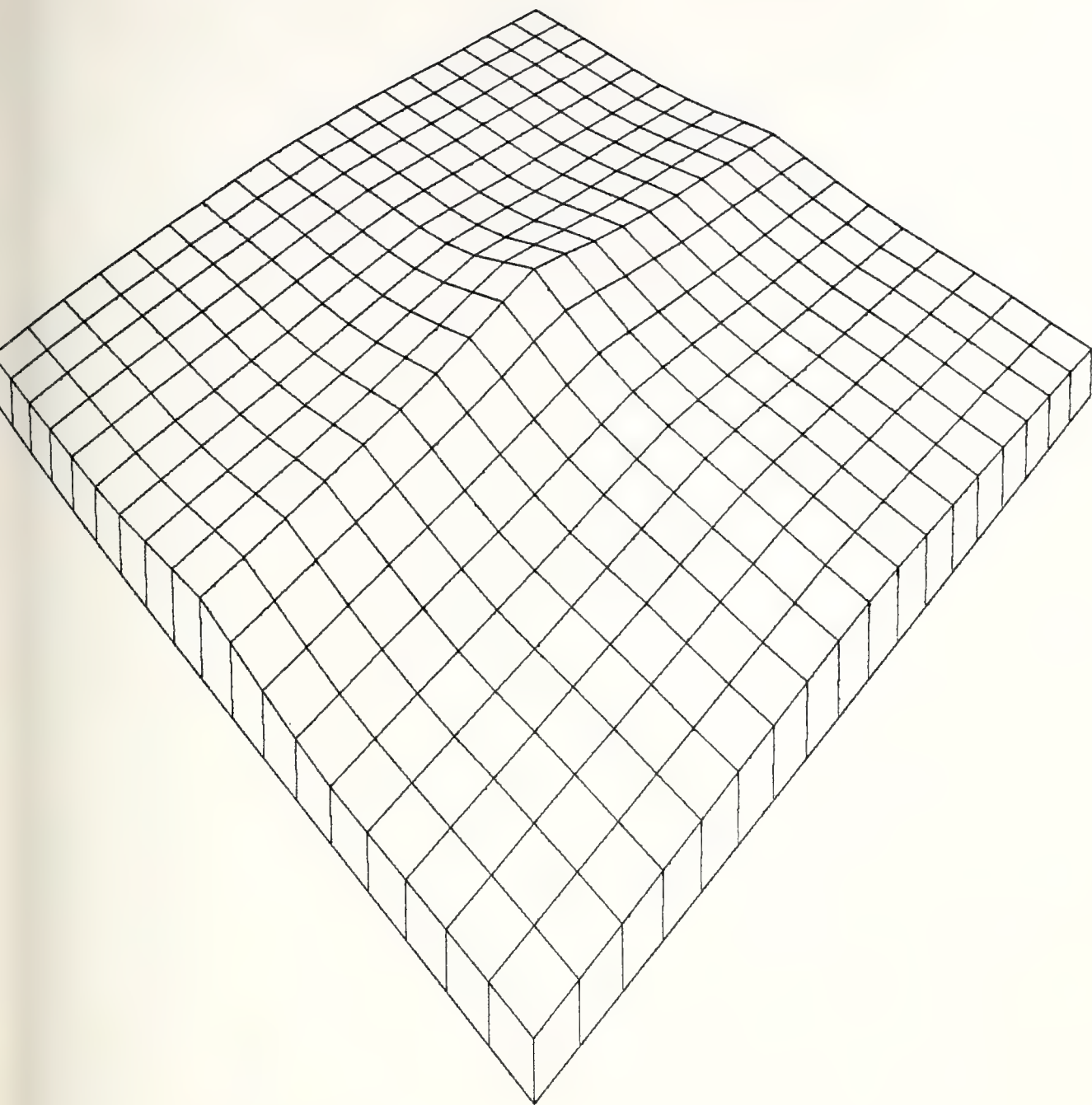


Figure 2.15
Measured Autocorrelation of Picture B

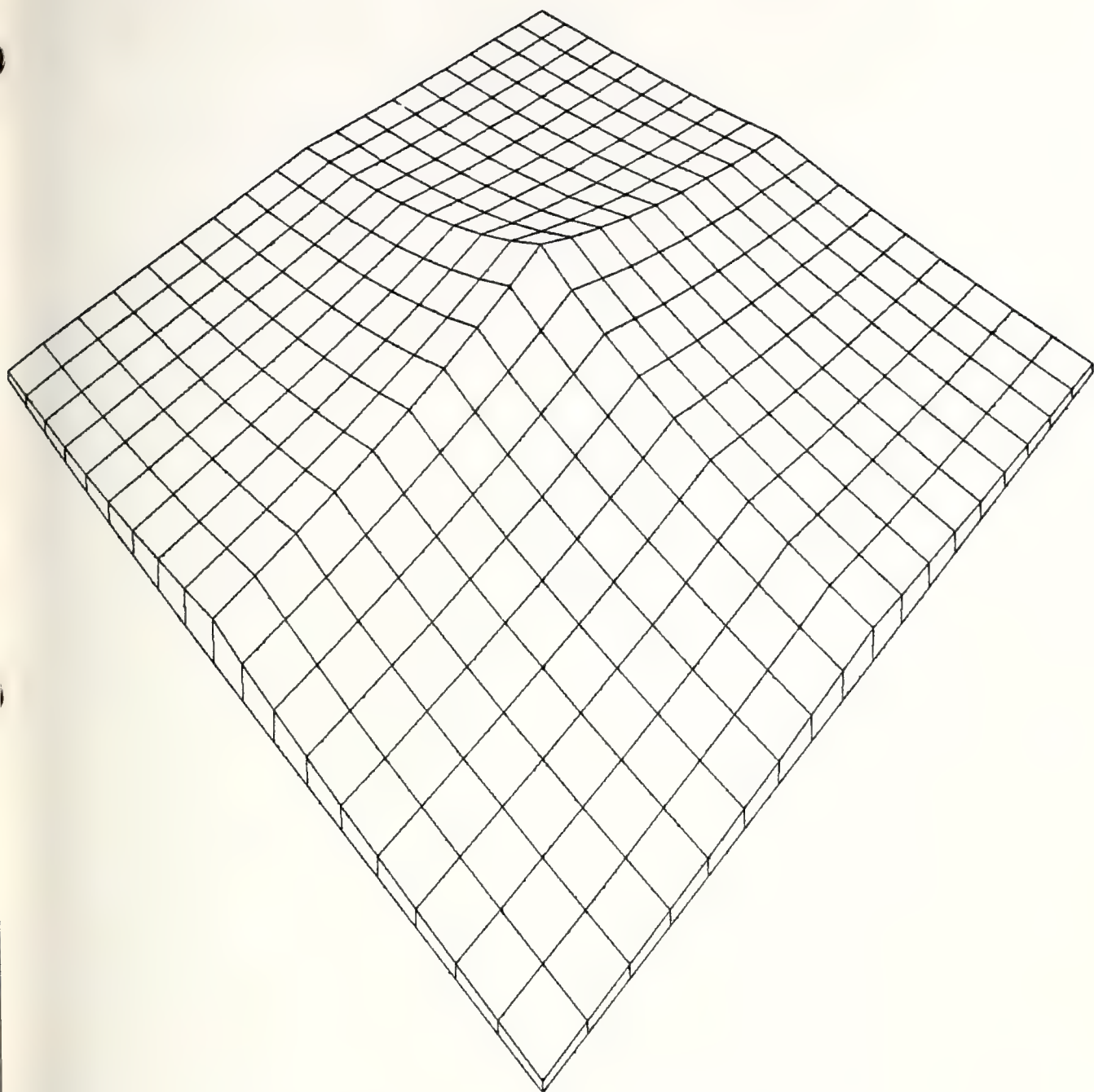


Figure 2.16
First Order Model of Picture B

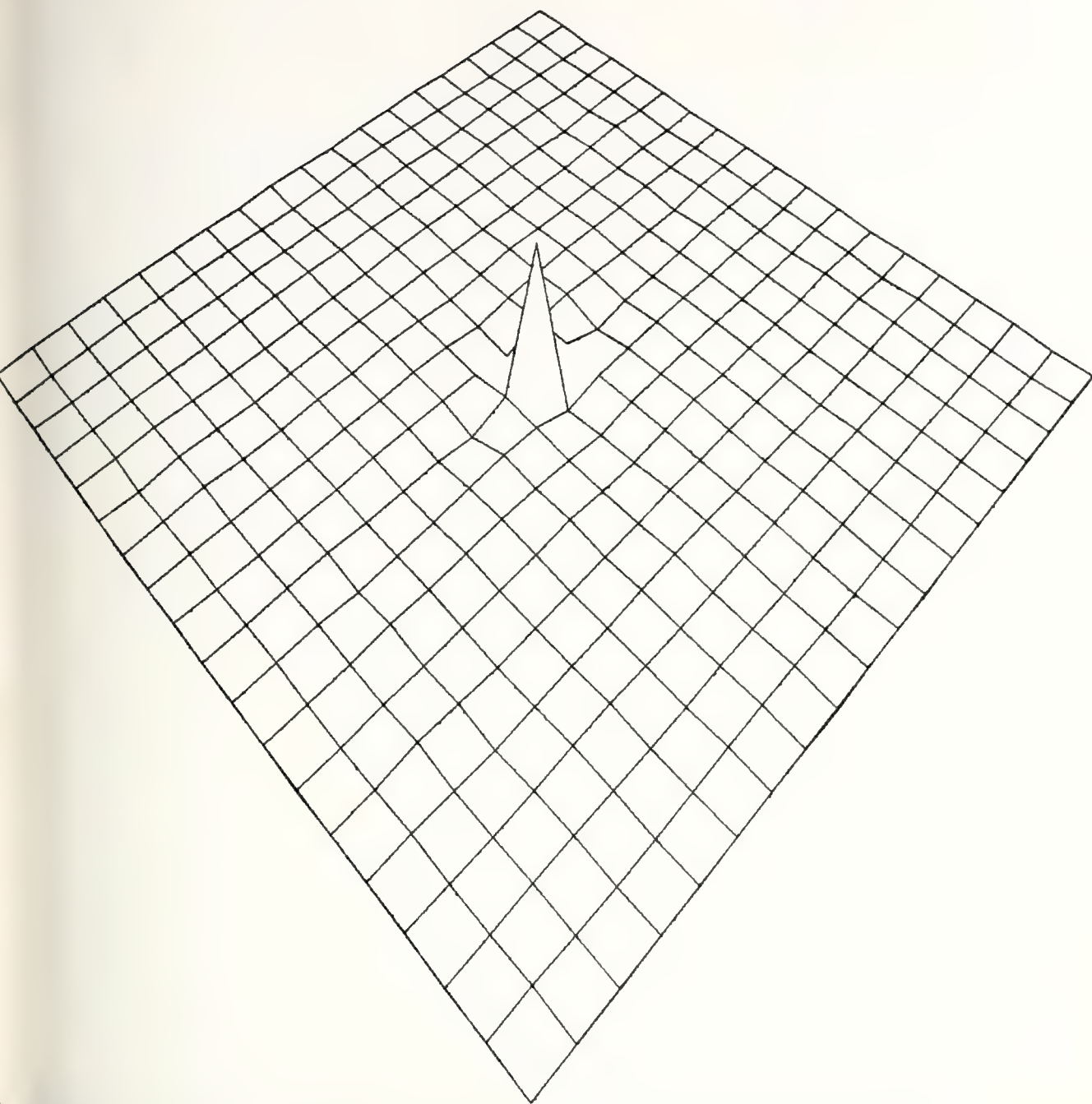


Figure 2.17

First Order Modeling Error of Picture B

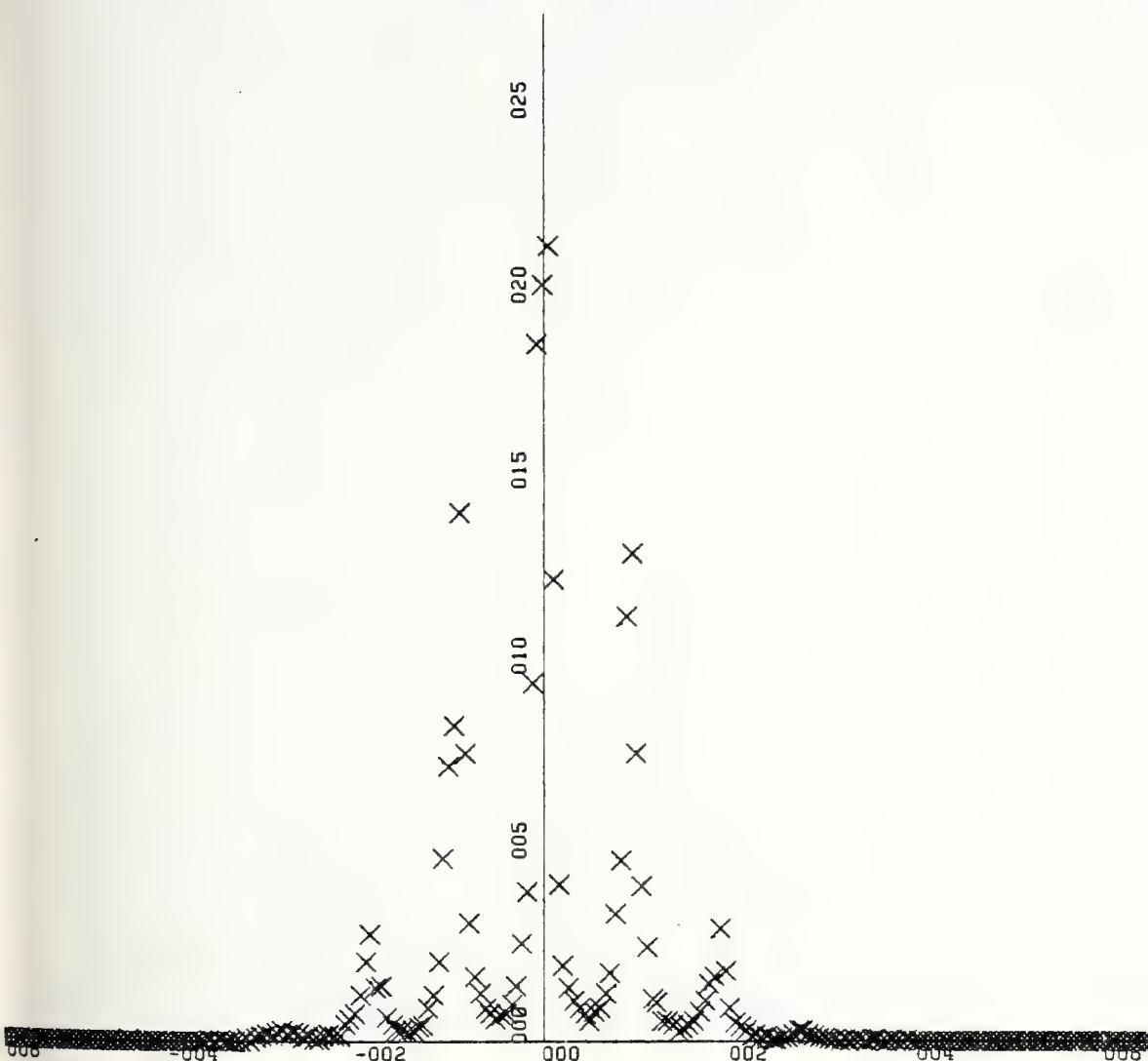


Figure 2.18
Histogram of First Order Modeling
Error of Picture B

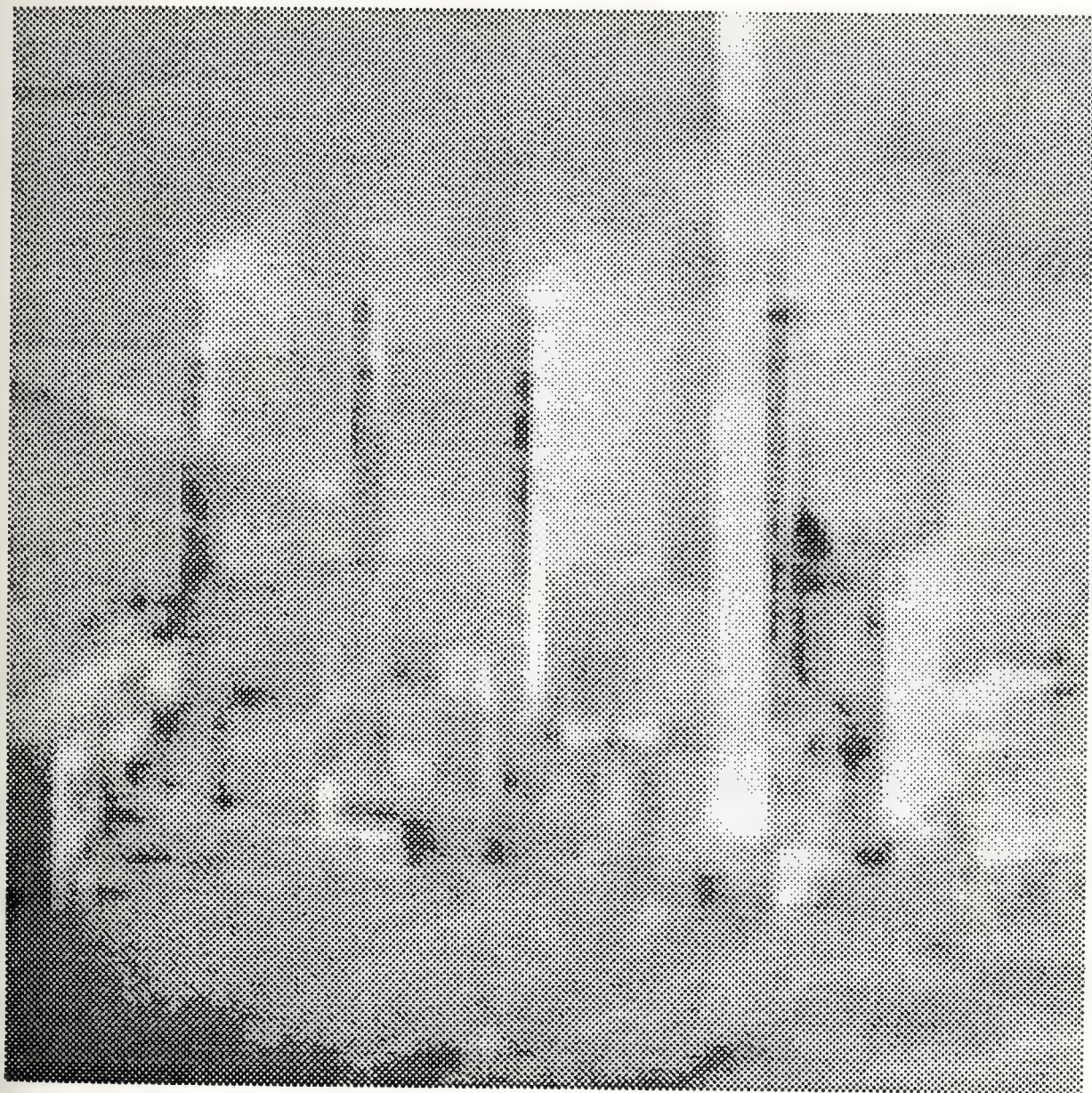


Figure 2.19
Reconstruction of Picture B with
Bandwidth Compressed 4:1

III. TWO-DIMENSIONAL RECURSIVE FILTERS

A. INTRODUCTION

A wide-sense Markov (WSM) sequence, corrupted by additive white noise, can be optimally (in the linear least squares sense) estimated by a recursive estimator. The fact that image intensity can be modeled by a generalization of a WSM sequence (Chapter II) motivates one to search for a two-dimensional Kalman estimator to smooth additive white observation noise in images. In references [3], [4] and [5] three filters are proposed. These filters use the dynamic model in equation (3.2), that is a particular case (first order WSM) of the more general model proposed in Chapter II. The structure of these filters is also the same (see equation 3.4), the difference being the method of gain computation. The reason of the existence of three Bayesian filters with the same structure, but different gains, is that they are all sub-optimum. It has been determined [10] that the optimality of the Kalman filter is not preserved when generalized to two dimensions, with such dynamic models. In reference [9] is proposed a vector model, described by a first-order linear n -dimensional vector difference equation, that is recursive in one parameter and generates the same random field used in [3-5]. With such a model the Kalman filter is applied preserving, of course, optimality. Unfortunately, the complexity of

implementation of such a filter is enormous, due to the size of the state vector, that is the number of pixels in one row, say, 256 or even 100. Also, its complexity increases drastically if we need to extend it to three-dimensional random fields, as in the case of time-frame, which will be addressed in chapter IV. Similarly, the extension to higher order WSM random fields increases its complexity.

Because of the above considerations, we will examine more carefully the sub-optimum filters in [3-5], due to their simplicity and adequacy to accommodate an extra dimension (time) or a higher order WSM random fields.

In this chapter we will compare the three filters [3-5] among themselves, as well as against the optimum non-recursive interpolator, constrained to the same data set. We will also introduce a recursive filter that is essentially the same as [5], but the computation of gains is accomplished without approximation. The figure of merit will be the error variance, since this is the cost function used by all three filters. Observe that filters [4] and [5] compute error variance in order to calculate the gains, but there is an approximation in the recursive equation, and this may result in big errors in the gains, as well as in the computed error variance.

In the remainder of this chapter, we will develop an algorithm to compute the variance of the estimation error, in order to compare the filters having the structure of equation (3.4) and dynamic model of equation (3.2). Since

filters [4] and [5] have similar algorithms for computation of gains, and, because filter [4] is clearly inferior to [5], only the former will be used in the comparison.

B. ALGORITHM FOR COMPUTATION OF ERROR VARIANCE

Given the dynamic model, the filter structure and the gain, we are going to develop an algorithm to compute the variance of the estimation error.

1. Dynamic Model

The picture is modeled by a zero-mean random field, homogeneous, with autocorrelation function:

$$R(i,j) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \quad (3.1)$$

The dynamic model is:

$$X(m,n) = \rho_v X(m-1,n) + \rho_h X(m,n-1) - \rho_v \rho_h X(m-1,n-1) + W(m,n) \quad (3.2)$$

where the gray level $X(m,n)$ has the autocorrelation of equation (3.1) and $W(m,n)$ is the random forcing input (or modeling error). $W(m,n)$ is zero-mean white noise, uncorrelated with $X(p,q)$, for all pixels (p,q) in region $X_{m,n}$ defined in figure 3.1.

$$E(W^2(m,n)) = \sigma^2 (1-\rho_v^2) (1-\rho_h^2) = Q \quad (3.3)$$

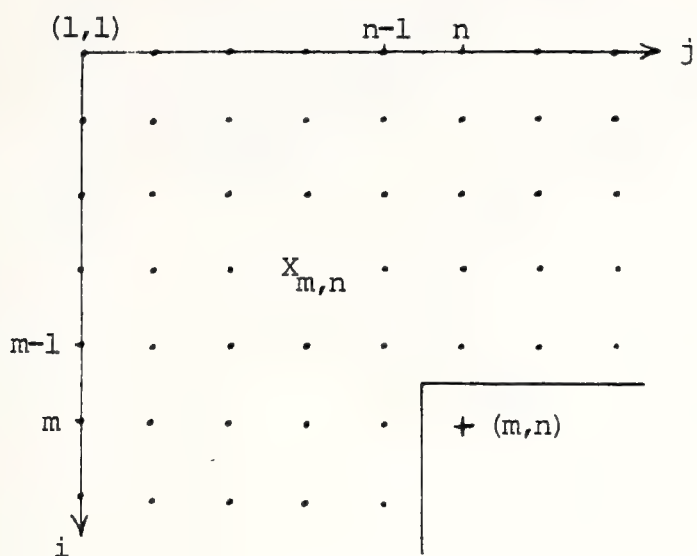


Figure 3.1

Definition of the region $X_{m,n}$

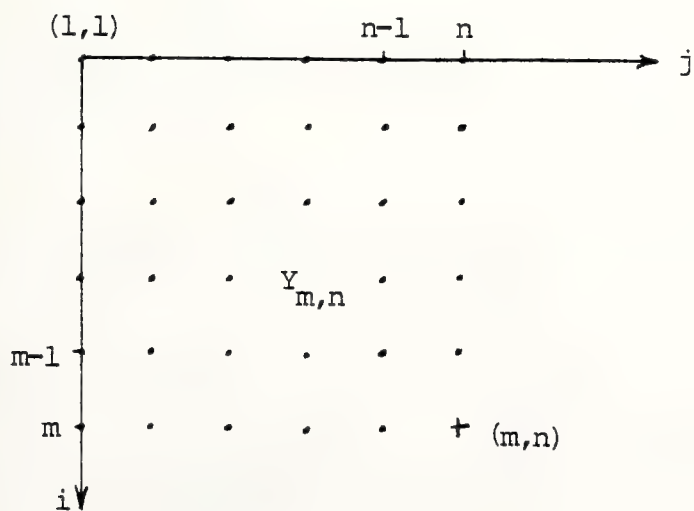


Figure 3.2

Definition of the region $Y_{m,n}$

2. Filter Structure

The filters [3,5] use equation:

$$\begin{aligned}\hat{X}(m,n) = & [1-K(m,n)] [\rho_v \hat{X}(m-1,n) + \rho_h \hat{X}(m,n-1) \\ & - \rho_v \rho_h \hat{X}(m-1,n-1)] + K(m,n) y(m,n)\end{aligned}\quad (3.4)$$

where $y(m,n)$ is the measurement of $X(m,n)$:

$$y(m,n) = X(m,n) + v(m,n) \quad (3.5)$$

where $v(m,n)$ is zero-mean additive white measurement noise, uncorrelated with $X(m,n)$, and having variance:

$$E(v^2(m,n)) = r$$

The estimate $\hat{X}(m,n)$ can be seen as the optimal linear combination of the prediction $\hat{X}_p(m,n)$, and the measurement $y(m,n)$:

$$\hat{X}(m,n) = [1-K(m,n)] \hat{X}_p(m,n) + K(m,n) y(m,n) \quad (3.6)$$

where:

$$\hat{X}_p(m,n) = \rho_v \hat{X}(m-1,n) + \rho_h \hat{X}(m,n-1) - \rho_v \rho_h \hat{X}(m-1,n-1) \quad (3.7)$$

The prediction $\hat{X}_p(m,n)$ carries all the information about $X(m,n)$, contained in the "past" measurements. The gain $K(m,n)$ is computed such that all the information about $X(m,n)$, contained in \hat{X}_p (or the "past" measurements) and $y(m,n)$, is effectively used. If that happens, the estimation error

$$e(m,n) = X(m,n) - \hat{X}(m,n)$$

is uncorrelated with the measurements used to estimate $X(m,n)$.

It is shown [10] that the estimation error can not be orthogonal to all the "past" measurements, but may be only to the region $Y_{m-1,n-1}$, defined in figure 3.2, which excludes row m and column n.

As a matter of fact, filters [3-5] do not have estimation error orthogonal to all measurements in $Y_{m-1,n-1}$ although incorrectly stated in [3] and [4]. A simple way to prove this above is using a counterexample. Let's examine the estimation of $X(2,3)$. According to [3] and [4], the estimation error of this pixel should be orthogonal to $y(1,2)$. Also the error in the estimation of $X(2,2)$ can't be orthogonal to $y(1,2)$, according to [10]. We are going to verify that these statements are incompatible.

The error in the estimation of $x(2,3)$ is

$$e(2,3) = x(2,3) - \hat{x}(2,3)$$

Using equations (3.2), (3.4) and the fact that $w(2,3)$ and $v(2,3)$ are uncorrelated with $y(1,2)$, it can be verified that

$$\begin{aligned} \overline{e(2,3)y(1,2)} &= [1-K(2,3) [\rho_v \overline{e(1,3)y(1,2)} \\ &\quad + \rho_h \overline{e(2,2)y(1,2)} - \rho_v \rho_h \overline{e(1,2)y(1,2)}]] \end{aligned}$$

Since the filters reduce to the one-dimensional Kalman filter for the first row:

$$\overline{e(1,3)y(1,2)} = \overline{e(1,2)y(1,2)} = 0$$

Thus

$$\overline{e(2,3)y(1,2)} = [1-K(2,3)] \rho_h \overline{e(2,2)y(1,2)}$$

Therefore, since in general $K(2,3) \neq 1$ and $e(2,2)$ can not be orthogonal to $y(1,2)$, we conclude that $e(2,3)$ is not orthogonal to $y(1,2)$.

3. Variance of the Estimation Error

In the following derivation we will simplify notation by dropping the argument (m,n) , where this does not cause confusion.

The variance of the estimation error is given by:

$$P(m,n) = E[(X-\hat{X})^2] = E(X^2) + E(\hat{X}^2) - 2E(X\hat{X})$$

Let's call:

$$E(\hat{X}^2) = \sigma_e^2$$

$$E(\hat{X}\hat{X}) = b$$

$$P(m,n) = \sigma^2 + \sigma_e^2 - 2b \quad (3.8)$$

a. Variance of the Estimate

To compute σ_e^2 , observe that \hat{X} can be written as a linear combination of all measurements in region $Y_{m,n}$ (see figure 3.2).

$$\begin{aligned} X(m,n) = & b_{11}Y(1,1) + \dots b_{1q}Y(1,q) + \dots b_{1n}Y(1,n) + \\ & b_{p1}Y(p,1) + \dots b_{pq}Y(p,q) + \dots b_{pn}Y(p,n) + \\ & b_{m1}Y(m,1) + \dots b_{mq}Y(m,q) + \dots b_{mn}Y(m,n) \end{aligned} \quad (3.9)$$

Define the column vectors:

$$\underline{B}^T = [b_{11} \quad b_{12} \quad \dots \quad b_{p1} \quad \dots \quad b_{mn}]$$

$$\underline{Y}^T = [y(1,1) \quad y(1,2) \quad \dots \quad y(p,1) \quad \dots \quad y(m,n)]$$

Substituting in (3.9):

$$\hat{X}(m,n) = \underline{B}^T \underline{Y} \quad (3.10)$$

The estimate variance is:

$$\sigma_e^2 = E(\hat{X}^2) = \underline{B}^T E(\underline{Y}\underline{Y}^T) \underline{B}$$

or

$$\sigma_e^2 = \underline{B}^T R_Y \underline{B} \quad (3.11)$$

where:

$$R_Y = E(\underline{Y}\underline{Y}^T)$$

Using equations (3.1) and (3.5), it can be easily seen that the elements of the autocorrelation matrix R_Y are given by:

$$R_Y(i,j) = \begin{cases} R(k-p, \ell-q) & \dots\dots (k, \ell) \neq (p, q) \\ \sigma^2 + r & \dots\dots\dots (k, \ell) = (p, q) \end{cases} \quad (3.12)$$

where i is the sequential number for the pairs (k, ℓ) and j for (p, q) as follows

$$(k, \ell) \text{ and } (p, q) = (1, 1), (1, 2), \dots, (m, n)$$

$$i \text{ and } j = 1, 2, \dots, mn$$

b. Covariance of X and \hat{X}

Using equation (3.10), the covariance b is given by:

$$b = E(X\hat{X}) = \underline{B}^T E(X\underline{Y})$$

or

$$b = \underline{B}^T R_{XY} \quad (3.13)$$

where

$$R_{XY} = E(X\underline{Y})$$

The elements of the vector R_{XY} are given by:

$$R_{XY}(i) = R(m-k, n-\ell) \quad (3.14)$$

where i is the sequential number for the pairs (k, ℓ) with

$$(k, \ell) = (1, 1), (1, 2), \dots, (m, n)$$

$$i = 1, 2, 3, \dots, mn$$

4. Calculation of the Coefficients

To find the vector of coefficients \underline{B} , observe in equation (3.9) that when:

$$Y(i,j) = \begin{cases} 1 & \dots\dots\dots (i,j) = (p,q) \\ 0 & \dots\dots\dots (i,j) \neq (p,q) \end{cases} \quad (3.15)$$

The result is:

$$\hat{X}(m,n) = b_{pq}$$

It can be easily verified that, using (3.15) and (3.4), all coefficients can be computed. By means of a simple computer program, therefore, we can compute the exact weight that any measurement has in the estimation of X .

Let's summarize the steps to compute the error variance at (m,n) :

(1) To find each coefficient $b_{p,q}$ use equation (3.4) with the values of y given by (3.15). The value of $\hat{X}(m,n)$ will be $b_{p,q}$. The initial conditions are:

$$\hat{X}(i,j) = 0 \quad \text{for } i = (p-1) \text{ or } j = (q-1)$$

(2) Use equations (3.11) and (3.12) to compute σ_e^2 .

(3) Use equations (3.13) and (3.14) to compute b .

(4) Compute error variance by (3.8).

C. ANOTHER RECURSIVE FILTER

In this section we will introduce another sub-optimum recursive filter having the same structure as those in [3-5]. It is similar to [5], but it computes the gains without approximations.

The objective is to find a way to calculate the gains, such that the variance of the estimation error is minimized.

Let's call the covariance, between X and its prediction \hat{X}_p , b_p ; and the variance of the prediction σ_p^2 .

Using equation (3.6):

$$b = E(X\hat{X}) = (1-K)E(X\hat{X}_p) + K E(Xy)$$

but:

$$b_p = E(X\hat{X}_p)$$

$$E(Xy) = E(X^2) = \sigma^2$$

Thus:

$$b = (1-K)b_p + K\sigma^2 \quad (3.16)$$

Squaring equation (3.6) and taking the expected value:

$$E(\hat{X}^2) = (1-K)^2 E(\hat{X}_p^2) + K^2 E(y^2) + 2K(1-K)E(y\hat{X}_p)$$

but:

$$\sigma_p^2 = E(\hat{X}_p^2)$$

$$E(y^2) = \sigma^2 + r$$

$$E(y\hat{X}_p) = E(X\hat{X}_p) = b_p$$

Therefore:

$$\sigma_e^2 = (1-K)^2 \sigma_p^2 + K^2 (\sigma^2 + r) + 2K(1-K)b_p \quad (3.17)$$

Substituting equations (3.16) and (3.17) in (3.8):

$$P(m,n) = (1-K)^2 (\sigma^2 + \sigma_p^2 - 2b_p) + K^2 r \quad (3.18)$$

Observe in (3.18) that σ_p^2 and b_p are independent of the gain $K(m,n)$, but they are functions of "past" gains. Therefore we can find $K(m,n)$, as a function of σ_p^2 , b_p , σ^2 and r , such that the error variance is minimized.

Differentiating (3.18) with respect to K and equating to zero:

$$K(m,n) = \frac{\sigma^2 + \sigma_p^2 - 2b_p}{\sigma^2 + \sigma_p^2 - 2b_p + r} \quad (3.19)$$

The variance of the prediction error is:

$$P_p(m,n) = E[(X - \hat{X}_p)^2] = \sigma^2 + \sigma_p^2 - 2b_p \quad (3.20)$$

Substituting in (3.19):

$$K(m,n) = \frac{P_p}{P_p + r} \quad (3.21)$$

Equation (3.20) is intuitively appealing, because it is the same as that of the one dimension scalar Kalman filter.

Substituting (3.19) into (3.18):

$$P_{MIN} = Kr \quad (3.22)$$

Therefore, with the gain calculated by (3.21), the minimum error variance is given by (3.22).

Observe that $P_p(m,n)$ is independent of $K(m,n)$, and, therefore, can be computed using "past" gains. Its calculation is quite similar to that of $P(m,n)$.

$$\hat{X}_p(m,n) = \underline{A}^T \underline{Y}_p \quad (3.23)$$

where:

$$\underline{A}^T = [a_{11} \quad a_{12} \quad \dots \quad a_{p1} \quad \dots \quad a_{m,n-1}]$$

$$\underline{Y}_p^T = [y(1,1) \quad y(1,2) \quad \dots \quad y(p,1) \quad \dots \quad y(m,n-1)].$$

The variance of \hat{X}_p is:

$$\sigma_p^2 = \underline{A}^T R_{Y_p} \underline{A} \quad (3.24)$$

where R_{Y_p} is the autocorrelation matrix of the measurements \underline{Y}_p .

The covariance $b_p(m,n)$, similarly to $b(m,n)$, is given by:

$$b_p = \underline{A}^T R_{XY_p} \quad (3.25)$$

where the vector R_{XY_p} is the correlation between X and \underline{Y}_p .

The coefficients \underline{A} are calculated identically to \underline{B} , but using the predictor equation (3.7) instead of (3.4).

A summary of gain calculation follows:

(a) Use equations (3.15) and (3.7) to find the coefficients \underline{A} . The initial conditions for (3.7) are:

$$\hat{X}(i,j) = 0 \quad \text{for } i = (p-1) \quad \text{and } j = (q-1).$$

(b) Use equation (3.24) to find σ_p^2 .

(c) Use equation (3.25) to find b_p .

(d) Use equation (3.20) to find P_p , and compute $K(m,n)$ by (3.21).

D. NON-RECURSIVE FILTER

The optimum recursive filter must have the same performance of the non-recursive Wiener filter, provided

the data set is the same. Since two-dimensional recursive filters, like those in [3-5], are not optimum, we want to find how good they are.

The minimum-mean-square-error (MMSE) in the estimation of $X(m,n)$, using the measurements in region $Y_{m,n}$ (see figure 3.2), is computed as follows.

Define the vector

$$\underline{Y} = \begin{bmatrix} y(1,1) \\ \vdots \\ y(1,n) \\ y(2,1) \\ \vdots \\ y(m,n) \end{bmatrix} \quad \underline{H} = \begin{bmatrix} h_{1,1} \\ \vdots \\ h_{1,n} \\ h_{2,1} \\ \vdots \\ h_{m,n} \end{bmatrix}$$

The estimate of $X(m,n)$ is

$$\hat{X}(m,n) = \underline{H}^T \underline{Y}$$

and the MSE

$$P(m,n) = E[(X - \hat{X})^2] = E[(X - \underline{H}^T \underline{Y})^2]$$

Let:

$$E(X^2) = \sigma^2$$

$$E(\underline{Y}\underline{Y}^T) = R_{YY}$$

$$E(X\underline{Y}) = R_{XY}$$

$$P(m,n) = \sigma^2 - 2\underline{H}^T R_{XY} + \underline{H}^T R_{YY} \underline{H} \quad (3.26)$$

Differentiate (3.26) and equate to zero:

$$\underline{H} = R_{YY}^{-1} R_{XY} \quad (3.27)$$

Substitute (3.27) into (3.26):

$$P_{MIN}(m,n) = \sigma^2 - \underline{H}^T R_{XY} \quad (3.28)$$

It can be shown that when the observation follows (3.5) the MMSE is given by

$$P_{MIN}(m,n) = h_{m,n} r \quad (3.29)$$

E. PERFORMANCE OF THE FILTERS

In this section we present the results of the comparison between filters [3], [5] and the one introduced in Section C, as well as the optimum interpolator of Section D.

First of all, we have compared filters [5] and that of Section C, since they are essentially the same, but the former uses an approximation. This comparison was accomplished by computing the gain and error variance of both filters for several situations. The result was that they presented practically the same values; therefore

we concluded that the approximation made in [5] is very good. This conclusion is important, because it allows the use of simple recursive equations, developed in [5], for gain computation.

The comparison of filters [3] and [5] against the optimum estimator is presented in table 3.1 and figure 3.3. In these results the coefficient of correlation is 0.9, in both directions, and the variance of X is 1.0. The variance of noise is in the range 0.01-1.0, therefore the signal-to-noise ratio goes from 0-20 dB.

From figure 3.3 and table 3.1, we see that filter [3] is better than [5], but both are quite close to the optimum filter. In the worst case, the error variance of [3] is 6.5% greater than the optimum, and [5] is 15%. Also, the error variance of [5] is 8% greater than that of filter [3]. It was observed, from other results, that [3] and [5] approach the optimum for lower coefficients of correlation and/or low noise.

Another experiment was performed to change the gain to see what MMSE can be achieved with such a filter structure. Figure 3.4 and table 3.2 show that filter [3] is quite close to the minimum error variance, although it is not exactly the minimum as can be verified in figure 3.5 and table 3.3. The conclusion is that filter [3] is better than [5], but it is not the best that can be done. As a matter of fact the present method of comparison can always be used to compute the best gain, although it is computationally complex.

The reason why filter [3] is better than [5] is that, although it assumes (incorrectly) that the estimation error is uncorrelated with the measurements in region $Y_{m-1,n-1}$ the gain is chosen such that the error is uncorrelated with $y(m,n)$. On the other hand, the error of filter [5] is not uncorrelated with $y(m,n)$, although it minimizes the up-dated error variance at each point. What happens is that the value of such minimum is not only dependent on the previous error variances, but also on the covariances of the "past" errors, since they are correlated.

The important conclusion is that the recursive filters [3], [5] and the one of section C, although not optimum, are quite close to optimality. Filter [3] is the best and simple enough to be extended to three dimensions, in order to exploit recursively the correlation in time. This will be done in the next chapter.

Table 3.1

| Noise Variance | Optimum | Filter [3] | Filter [5] |
|---|-----------|------------|------------|
| 0.01 | 0.0083729 | 0.0084204 | 0.0083936 |
| 0.04 | 0.0262313 | 0.0269059 | 0.0266656 |
| 0.09 | 0.0477314 | 0.0497128 | 0.0494879 |
| 0.16 | 0.0706634 | 0.0743120 | 0.0747814 |
| 0.25 | 0.0940204 | 0.0994702 | 0.1014409 |
| 0.36 | 0.1172805 | 0.1245305 | 0.1287582 |
| 0.49 | 0.1401542 | 0.1491169 | 0.1562232 |
| 0.64 | 0.1624769 | 0.1730206 | 0.1834437 |
| 0.81 | 0.1841581 | 0.1961401 | 0.2101098 |
| 1.00 | 0.2051533 | 0.2184254 | 0.2359806 |
| Picture Variance = 1.0 Picture Correlation = 0.9 | | | |

PICTURE VARIANCE = 1.0
 PICTURE CORRELATION = 0.9
 (o) - optimum filter
 (*) - filter [3]
 (+) - filter [5]

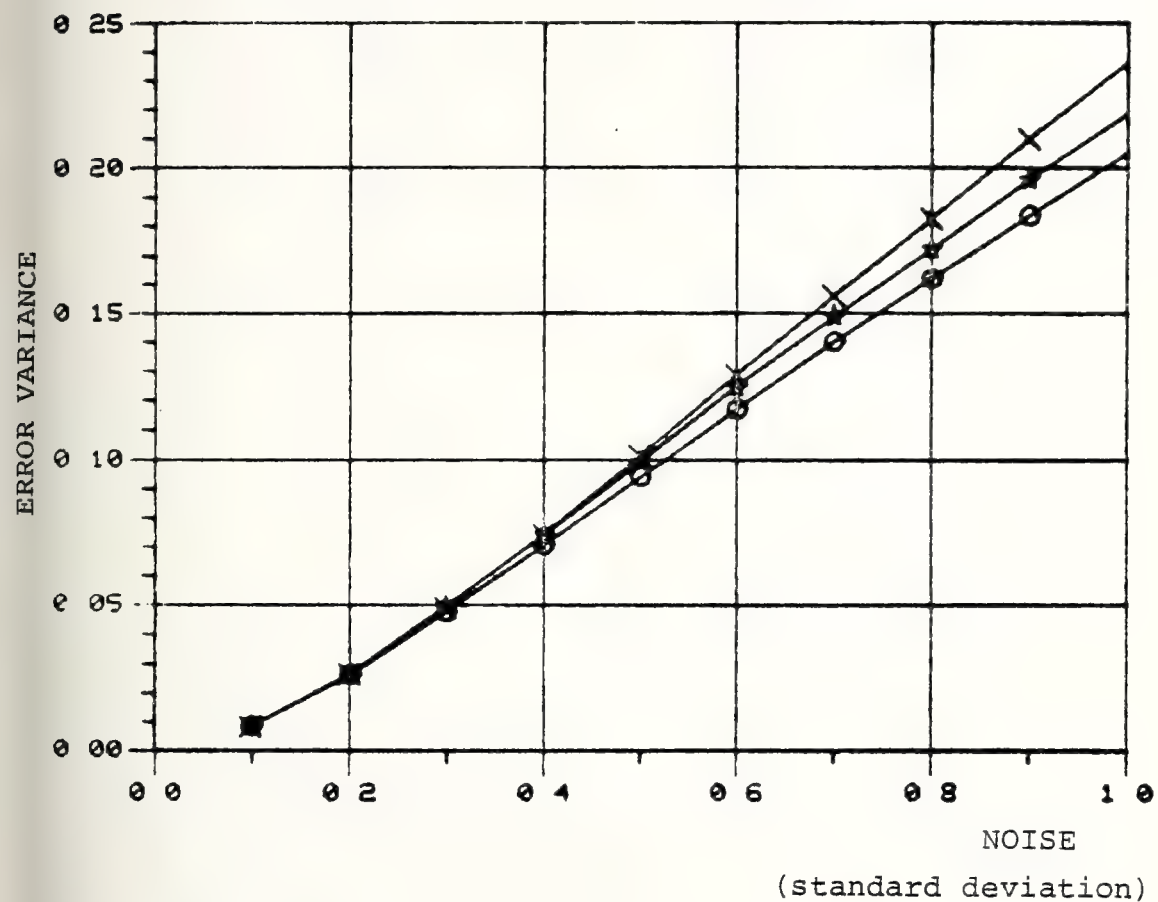


Figure 3.3
 Comparison of the Filters

Table 3.2

| Gain | Error Variance | |
|---|----------------|------------|
| 0.07982749 | 0.26742077 | |
| 0.09579301 | 0.24629241 | |
| 0.11175853 | 0.23272502 | |
| 0.12772405 | 0.22441578 | |
| 0.14368957 | 0.21995461 | |
| 0.15965503 | 0.21842539 | Filter [3] |
| 0.17562050 | 0.21918494 | |
| 0.19158608 | 0.22175616 | |
| 0.20755148 | 0.22577757 | |
| 0.22351706 | 0.23097461 | |
| 0.23948258 | 0.23713821 | Filter [5] |
| Picture Variance = 1.0 Noise Variance = 1.0 Picture Correlation = 0.9 | | |

PICTURE VARIANCE = 1.0
NOISE VARIANCE = 1.0
PICTURE CORRELATION = 0.9

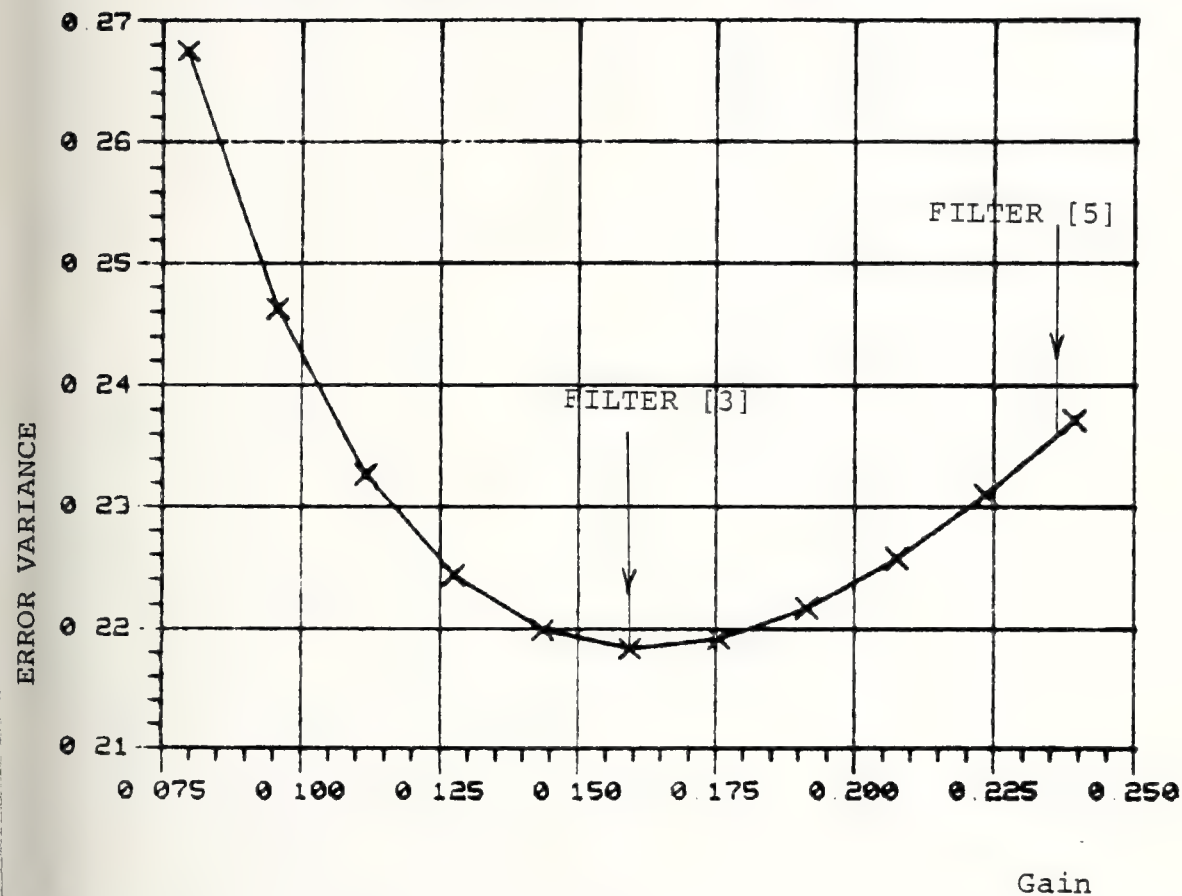


Figure 3.4
Error Variance vs Steady-State Gain

Table 3.3

| Gain | Error Variance | |
|----------------------------|----------------|------------|
| 0.29999995 | 0.09970987 | Filter [3] |
| 0.30999994 | 0.09923345 | |
| 0.31999993 | 0.09892100 | |
| 0.32999992 | 0.09875959 | |
| 0.33999991 | 0.09873748 | Best Gain |
| 0.34999990 | 0.09884429 | |
| 0.35999990 | 0.09907085 | |
| 0.36999995 | 0.09940886 | |
| 0.37999994 . | 0.09985095 | |
| 0.38999993 | 0.10039049 | |
| 0.39999992 | 0.10102153 | Filter [5] |
| Picture Variance = 1.00 | | |
| Noise Variance = 0.25 | | |
| Picture Correlation = 0.90 | | |

PICTURE VARIANCE = 1.00
NOISE VARIANCE = 0.25
PICTURE CORRELATION = 0.90

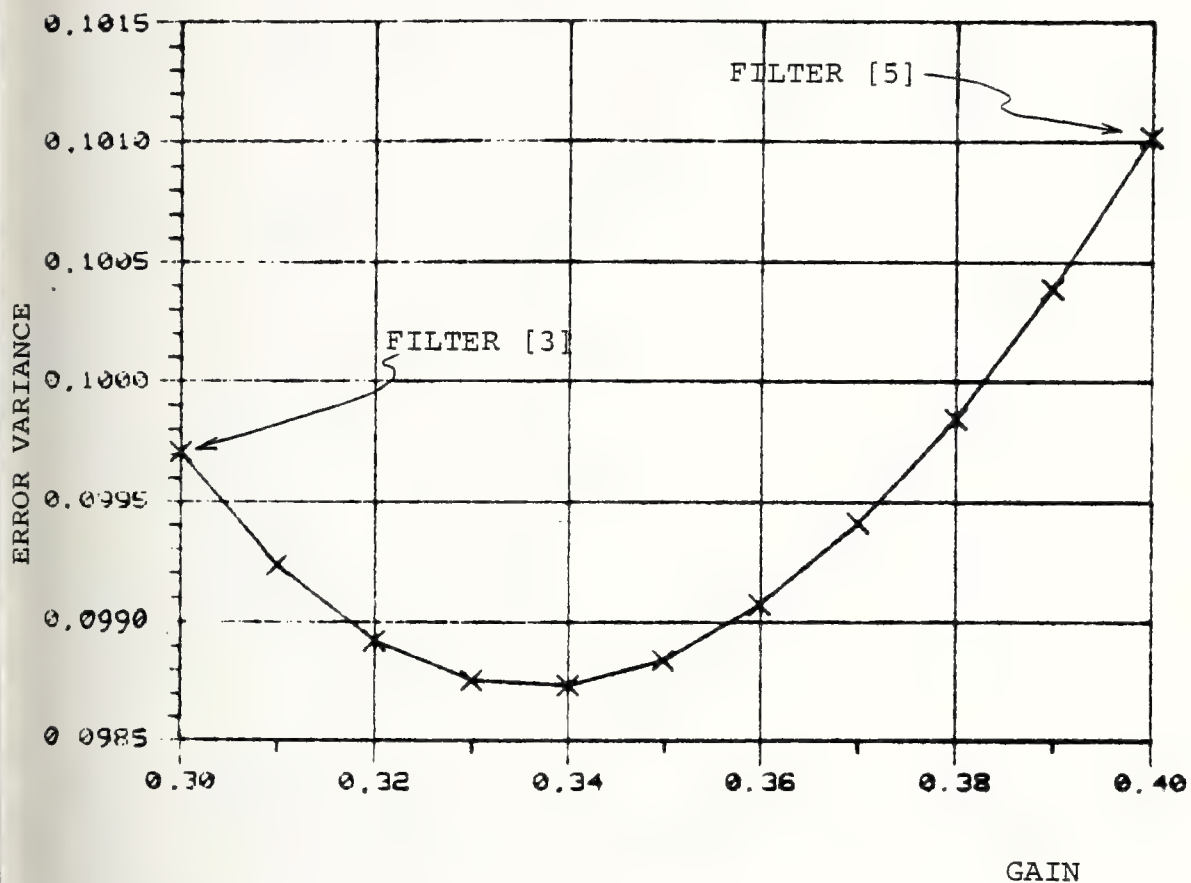


Figure 3.5
Error Variance vs Steady-State Gain

IV. THREE-DIMENSIONAL RECURSIVE FILTER

A. INTRODUCTION

The two-dimensional recursive filters, presented in Chapter III, use only the spatial correlation between pixels in order to estimate the gray levels of noisy pictures. In this chapter we introduce a recursive filter that takes advantage of both correlations in space and in time, when one has a group of pictures sequenced in time. Experimental evidence, presented in [1], shows that television pictures are correlated in time. This is also an intuitive result, since knowing one frame we often can guess the next.

The filter developed here is an extension of the two-dimensional recursive filter [3], since it is the best, according to the results of Chapter III. In any case, its performance will be analyzed, in order to evaluate the improvement resultant in using the time correlation.

B. FILTER DESIGN

The filter will be developed under the following conditions:

(a) The time-frame is modeled as an homogeneous random field with zero-mean (or known mean) and autocorrelation function $R(i,j,k)$, where i,j and k are the distances between pixels in the vertical, horizontal and time coordinates,

respectively. The autocorrelation is like the one introduced in Chapter II, with kernels given by equation (2.26).

(b) The dynamic model is given by the partial difference equation:

$$X(m,n,t) = \underline{A} \underline{X} + W(m,n,t) \quad (4.1)$$

where \underline{A} is a row vector of coefficients and \underline{X} is a column vector of adjacents. The modeling error $W(m,n,t)$ is white noise and uncorrelated with the X 's in region X_{mnt} (see figure 4.1).

In order to make clear the present derivation, let's assume a specific form for the autocorrelation, say, first order in all dimensions. The development remains valid for the k -order kernels given by equation (2.26).

The autocorrelation is:

$$R(i,j,k) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \rho_t^{|k|} \dots \quad i,j,k = 0, \pm 1, \pm 2, \dots \quad (4.2)$$

The coefficients are:

$$\underline{A} = [\rho_v \quad \rho_h \quad \rho_t \quad -\rho_v \rho_h \quad -\rho_v \rho_t \quad -\rho_h \rho_t \quad \rho_v \rho_h \rho_t] \quad (4.3)$$

The adjacents (see figure 4.2) are:

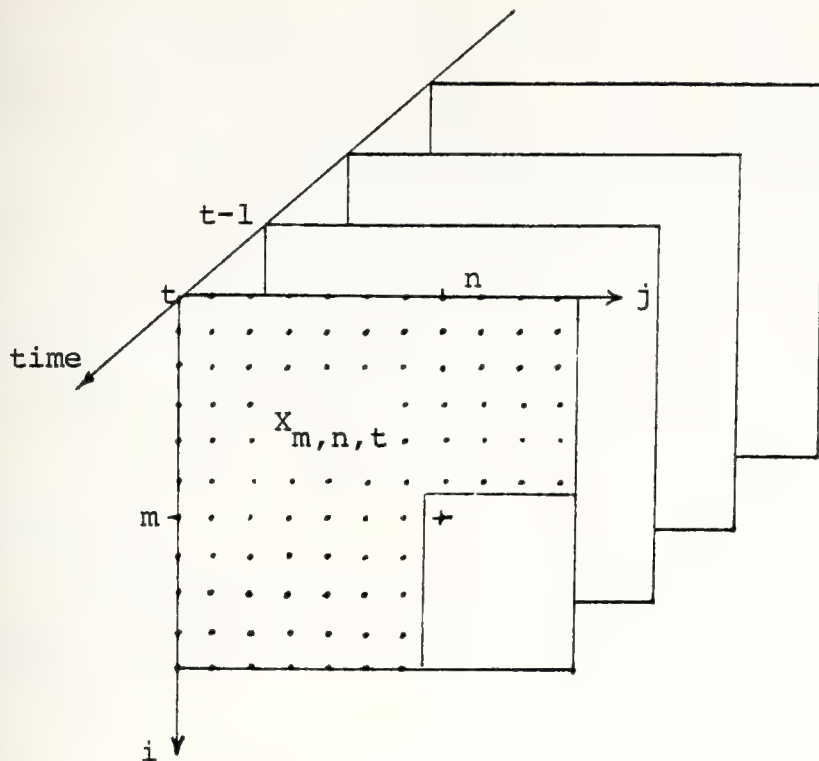


Figure 4.1

Definition of the region $X_{m,n,t}$: the region includes the past frames and the pixels shown

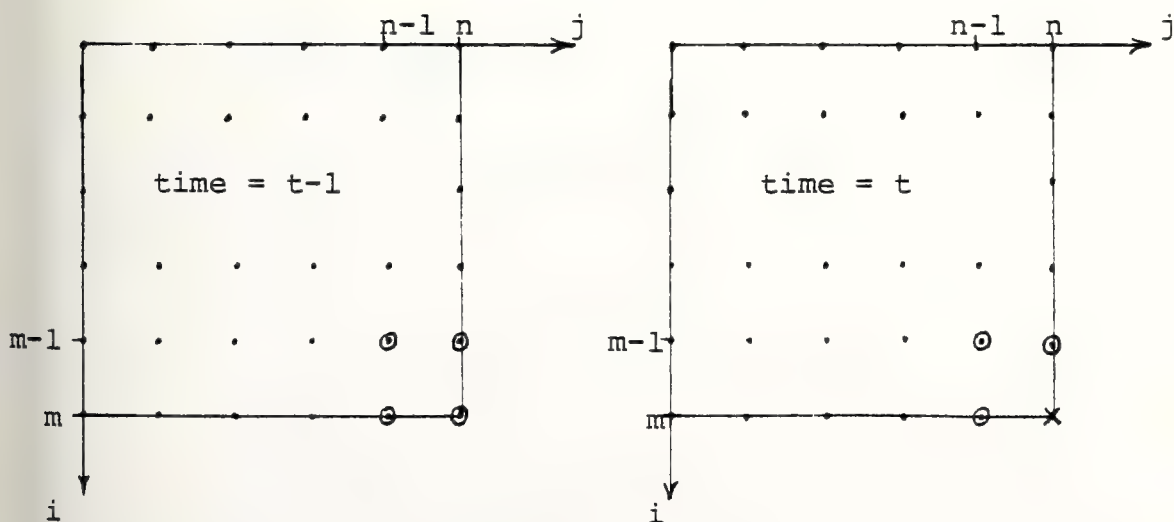


Figure 4.2

Configuration of the adjacents

$$\underline{X}^T = \begin{bmatrix} X(m-1,n,t) & X(m,n-1,t) & X(m,n,t-1) & X(m-1,n-1,t) \\ X(m-1,n,t-1) & X(m,n-1,t-1) & X(m-1,n-1,t-1) \end{bmatrix} \quad (4.4)$$

The variance of W is

$$E[W^2(m,n,t)] = Q = \sigma^2 (1-\rho_v^2) (1-\rho_h^2) (1-\rho_t^2) \quad (4.5)$$

(c) The pictures are contaminated with zero-mean white noise, uncorrelated with the X's.

$$y(m,n,t) = X(m,n,t) + v(m,n,t) \quad (4.6)$$

The noise variance is:

$$E(v^2(m,n,t)) = r$$

1. Filter Structure

Since we want to generate a random field X as close as possible to X, let's choose a structure similar to the dynamic model (4.1):

$$\hat{X}(m,n,t) = \underline{B} \hat{X} + K(m,n,t)y(m,n,t) \quad (4.7)$$

In what follows we will drop the argument (m,n,t), where it could not cause confusion.

Observe that \hat{X} can be written as a linear combination of all measurements within region Y_{mnt} , defined in figure 4.3. The optimum recursive filter must have the gains such that the weights of the measurements are exactly the same as for the non-recursive estimator, constrained to the same data set Y_{mnt} . The criteria of optimality is minimization of the Mean Square Error (MSE). It will be seen that it is impossible to have an optimum recursive filter, constrained to the dynamic model (4.1).

2. Orthogonality Principle

A necessary condition of optimality is that the estimation error be uncorrelated with the data set. Let's apply this condition to find the coefficients \underline{B} and K .

The estimation error is:

$$e(m,n,t) = X - \hat{X} \quad (4.8)$$

The orthogonality condition is:

$$E[e(m,n,t)y(p,q,r)] = 0 \quad (4.9)$$

for all (p,q,r) in Y_{mnt} (see figure 4.3)

First, let's apply (4.9) to the measurement $y(m,n,t)$:

$$\begin{aligned} E(e y) &= E[(X - \hat{X})y] = E[y(X - \underline{B}\hat{X} - Ky)] \\ &= E[Xy - Ky^2 - \underline{B}y\hat{X}] \end{aligned}$$

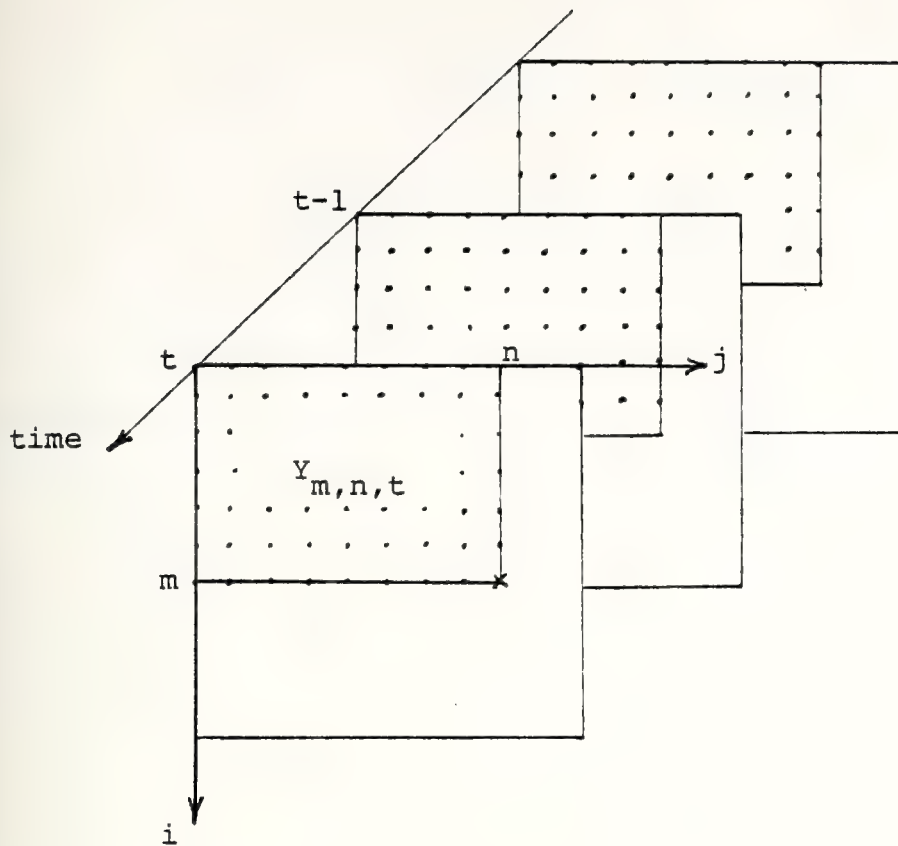


Figure 4.3
Definition of the region $Y_{m,n,t}$

Using equations (4.2) and (4.6):

$$E(ey) = (1-K)\sigma^2 - Kr - \underline{B}E(\hat{X}\underline{X}) = 0$$

$$K(\sigma^2 + r) + \underline{B}E(\hat{X}\underline{X}) = \sigma^2 \quad (4.10)$$

Let's apply (4.9) to the rest of the measurements.

Substituting (4.7) into (4.8):

$$\begin{aligned} e &= X - \hat{X} = X - Ky - \underline{B} \hat{X} \\ &= (1-K)X - Kv - \underline{B} \underline{X} + \underline{B} e \end{aligned}$$

$$e = (1-K) \left[X - \frac{1}{1-K} \underline{B} \underline{X} \right] - Kv + \underline{B} e \quad (4.11)$$

Substituting (4.11) into (4.9):

$$\underline{B} E[y(p,q,r)e] + (1-K)E\left[\left(X - \frac{1}{1-K} \underline{B} \underline{X}\right)y(p,q,r)\right] = 0 \quad (4.12)$$

for $y(p,q,r) \neq y(m,n,t)$, and in Y_{mnt} .

The problem now is to choose the coefficients \underline{B} and K , such that equations (4.10) and (4.12) are satisfied.

At this point we follow [3] by choosing

$$\underline{B} = (1-K)\underline{A} \quad (4.13)$$

With such a choice, the second term of (4.12) becomes:

$$(1-K)E[W(m,n,t)y(p,q,r)]$$

Where we have used (4.1), and this term vanishes because $W(m,n,t)$ is uncorrelated with all measurements in region X_{mnt} . Substituting (4.13) into (4.10):

$$K(m,n,t) = \frac{\sigma^2 + b_p}{\sigma^2 + b_p + r} \quad (4.14)$$

where we have defined:

$$\begin{aligned} \hat{X}_p(m,n,t) &= \underline{A} \underline{\hat{X}} \\ b_p(m,n,t) &= E(\underline{\hat{X}} \underline{\hat{X}}_p) \end{aligned} \quad (4.15)$$

Equation (4.15) seems to be an adequate choice for the first-step predictor, also (4.14) is an intuitive result for the gain, since it is unity for zero noise ($r=0$) and decreases when noise increases.

Unfortunately, the first term of (4.12) remains and we could not force it to vanish. Choosing K by (4.14) we have forced the estimation error to be uncorrelated with $y(m,n,t)$, but it is not correct to conclude, by induction, that the estimation errors of the adjacents (vector \underline{e}) are uncorrelated with the other measurements.

Take, for example, the error:

$$e(m,n,t-1) = X(m,n,t-1) - \hat{X}(m,n,t-1)$$

The estimate didn't use any measurements of frame t , therefore the error can not be uncorrelated with them.

As a matter of fact, the errors might only be uncorrelated with the measurements within region $Y_{m-1,n-1,t-1}$. In reference [3], for the two-dimensional case, it is stated that the error is uncorrelated with all measurements in a similar region. Unfortunately, that is not correct. The estimation error, in the two or three dimensional cases, is uncorrelated only with $y(m,n,t)$, because K was computed under such conditions, and with the measurement $y(1,1,1)$. This can be seen from (4.12) and (4.13):

$$E[ey(p,q,r)] = (1-K)E[\underline{A} \underline{e} y(p,q,r)] \quad (4.16)$$

$$\text{for } (p,q,r) \neq (m,n,t)$$

Initializing the filter with the mean value of X ($=0$), it can be verified, using (4.16), that $e(m,n,t)$ is uncorrelated with $y(1,1,1)$.

Although equation (4.14) is not the optimum choice for the gain, we have shown in Chapter III that it was the best in the two-dimensional case, therefore we hope it is also good in three dimensions. This will be verified at the end of this chapter.

Using (4.7) and (4.13), the filter equation becomes:

$$\hat{X}(m,n,t) = (1-K)\underline{A} \hat{X} + Ky \quad (4.17)$$

Alternatively, using (4.15):

$$\hat{X}(m,n,t) = (1-K)\hat{X}_p + Ky \quad (4.18)$$

3. Gain Computation

Let's develop a recursive equation to compute b_p and, therefore, the gain. Define a function:

$$F_{pqr}(m,n,t) = E[X(m,n,t)\hat{X}(p,q,r)]$$

Using (4.17):

$$F_{pqr} = [1-K(p,q,r)]\underline{A} \underline{F} + K(p,q,r)R(m-p,n-q,t-r) \quad (4.19)$$

where:

$$\underline{F}^T = [F_{p-1,q,r} \quad F_{p,q-1,r} \quad F_{p,q,r-1} \quad F_{p-1,q-1,r} \quad F_{p-1,q,r-1} \quad F_{p,q-1,r-1} \quad F_{p-1,q-1,r-1}] \quad (4.20)$$

Initializing with \hat{X} equal to the mean of X , outside the picture, the initial conditions for F are:

$$F_{p,q,r}(m,n,t) = 0 \quad \text{for } p,q \text{ or } r = 0$$

Now, using (4.15) and (4.19):

$$b_p(m,n,t) = \underline{A} \underline{F}_p \quad (4.21)$$

where \underline{F}_p is the value of \underline{F} for $(p,q,r) = (m,n,t)$.

Summary of gain calculation:

- (a) Use (4.19) to compute each component of the vector \underline{F}_p .
- (b) Use (4.21) to compute $b_p(m,n,t)$.
- (c) Compute $K(m,n,t)$ by (4.14)

Observe that the calculation of each gain $K(m,n,t)$ requires scanning the time-frame from $(1,1,1)$ through the pixel (m,n,t) . Fortunately, as will be seen, after a few frames the gain reaches steady state, therefore the computation can be reduced to the top left corner of a few frames.

C. PERFORMANCE OF THE FILTER

The method used in Chapter III, for the two dimensional case, can be easily extended to three dimensions. The variance of the estimation error is:

$$\begin{aligned} P(m,n,t) &= E[(X - \hat{X})^2] \\ &= E(X^2) + E(\hat{X}^2) - 2E(X\hat{X}) \end{aligned}$$

or:

$$P(m,n,t) = \sigma^2 + \sigma_e^2 - 2b \quad (4.22)$$

where:

$$E(X^2) = \sigma^2$$

$$E(\hat{X}^2) = \sigma_e^2$$

$$E(X\hat{X}) = b$$

The covariance b can be calculated using b_p and K :

$$b(m,n,t) = E(X\hat{X}) = (1-K)E(X\hat{X}_p) + KE(Xy)$$

$$b(m,n,t) = (1-K)b_p + K\sigma^2 \quad (4.23)$$

To compute the variance of the estimator, write \hat{X} as a linear combination of the measurements in region Y_{mnt} :

$$\hat{X}(m,n,t) = \underline{c}^T \underline{Y} \quad (4.24)$$

where \underline{c} is a column vector of coefficients, function of the gains, and \underline{Y} is a column vector containing all measurements in Y_{mnt} .

The variance of \hat{X} is:

$$\sigma_e^2 = c^T R_Y c \quad (4.25)$$

where R_Y is the correlation matrix of the vector \underline{Y} :

$$R_Y = E(\underline{Y}\underline{Y}^T)$$

The key, in this computation, is the method of finding the coefficients \underline{c} . This can be easily accomplished using the filter equation (4.17) with:

$$y(i,j,k) = \begin{cases} 1 & \dots\dots\dots (i,j,k) = (p,q,r) \\ 0 & \dots\dots\dots (i,j,k) \neq (p,q,r) \end{cases}$$

$$\begin{aligned} \hat{x}(i,j,k) &= 0 \quad \text{for } k = (r-1) \text{ or} \\ &\quad i = (p-1) \text{ or} \\ &\quad j = (q-1) \end{aligned}$$

Initializing (4.17) as above, and starting at (p,q,r) , the resultant value at (m,n,t) will be the coefficient of $y(p,q,r)$ in the estimate $\hat{x}(m,n,t)$. In this way all coefficients can be found and, therefore, the error variance.

We are also interested in the first-step predictor given by (4.15). Its error variance is:

$$P_p(m,n,t) = E[(X - \hat{X}_p)^2] \quad (4.26)$$

It can be related with $P(m,n,t)$ using (4.18) and (4.26):

$$\begin{aligned} X - \hat{X} &= X - (1-K)\hat{X}_p - Ky \\ &= (1-K)(X - \hat{X}_p) - Kv \end{aligned}$$

Squaring and taking expectations:

$$\begin{aligned} P &= (1-K)^2 P_p + K^2 r \\ P_p(m,n,t) &= \frac{1}{(1-K)^2} (P - K^2 r) \end{aligned} \quad (4.27)$$

D. RESULTS

In figures (4.4) through (4.9) some results of gain calculations are shown. For the first frame the gain is the same as in [3] for two dimensions. From these results we can observe that the gain reaches a steady-state value very fast, at about frame number 4. Also, for the same frame, the steady-state is reached at about pixel (4,4). That is an interesting result, since it simplifies substantially gain calculations. Also we can think of using a constant gain for the whole time-frame, or a few gains for the first 4 frames.

In tables 4.1 through 4.3 a comparison is shown between the three-dimensional filter, estimate and prediction, and the two-dimensional recursive filter. It can be seen in table 4.1 that exploitation of time correlation can be

quite advantageous in cases where time correlation is very high, compared with spatial correlations. In this example, the estimation error variance is reduced by about 33%, due to the use of time correlation. In table 4.2 a comparison with equal amounts of correlation in time and space is shown. In this example the improvement was about 10%. It can be seen in table 4.3 that exploitation of time correlation is not so advantageous when the images are more correlated in space than in time. In this example the improvement was only around 6%.

SIGNAL/NOISE = 10 dB

FRAME CORRELATION: $\rho_v = \rho_h = \rho_t = 0.9$

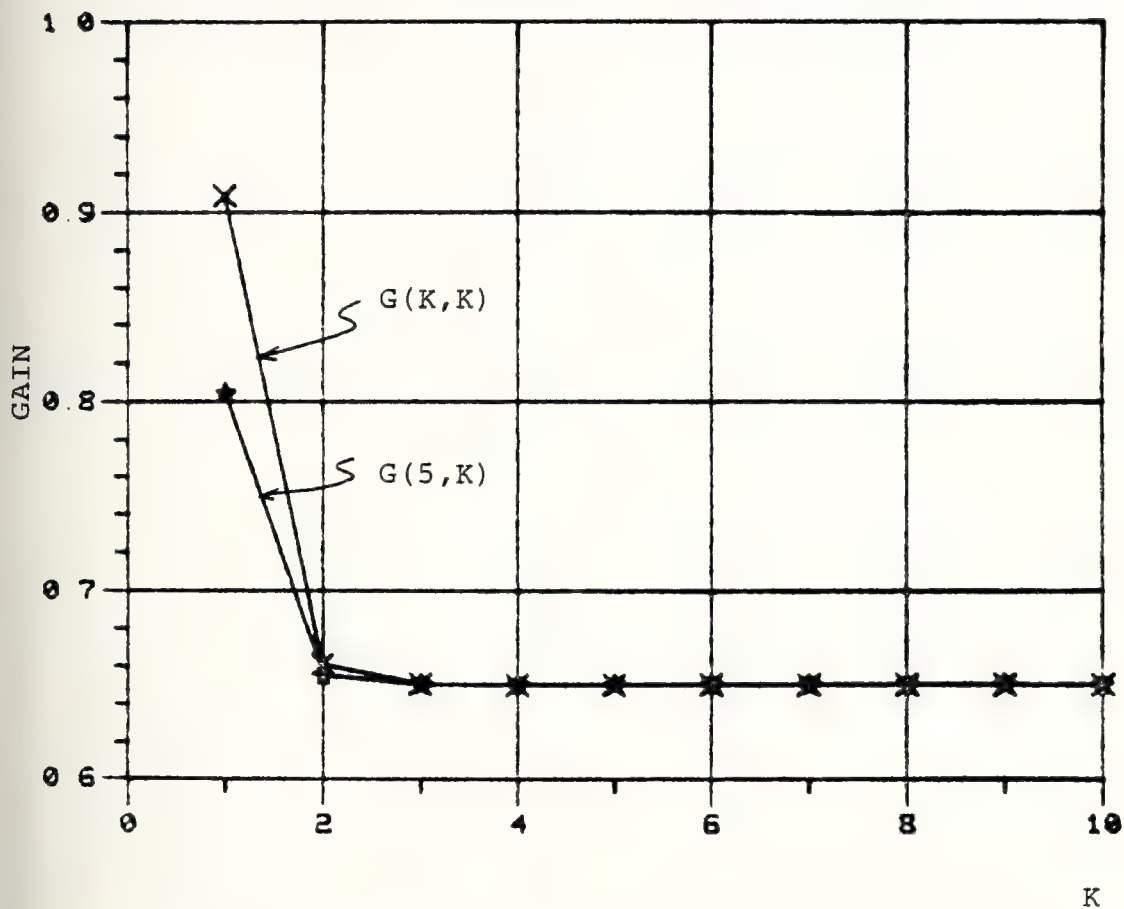


Figure 4.4
Gains for the First Frame

SIGNAL/NOISE = 10 dB

FRAMES CORRELATION: $\rho_v = \rho_h = \rho_t = 0.8$

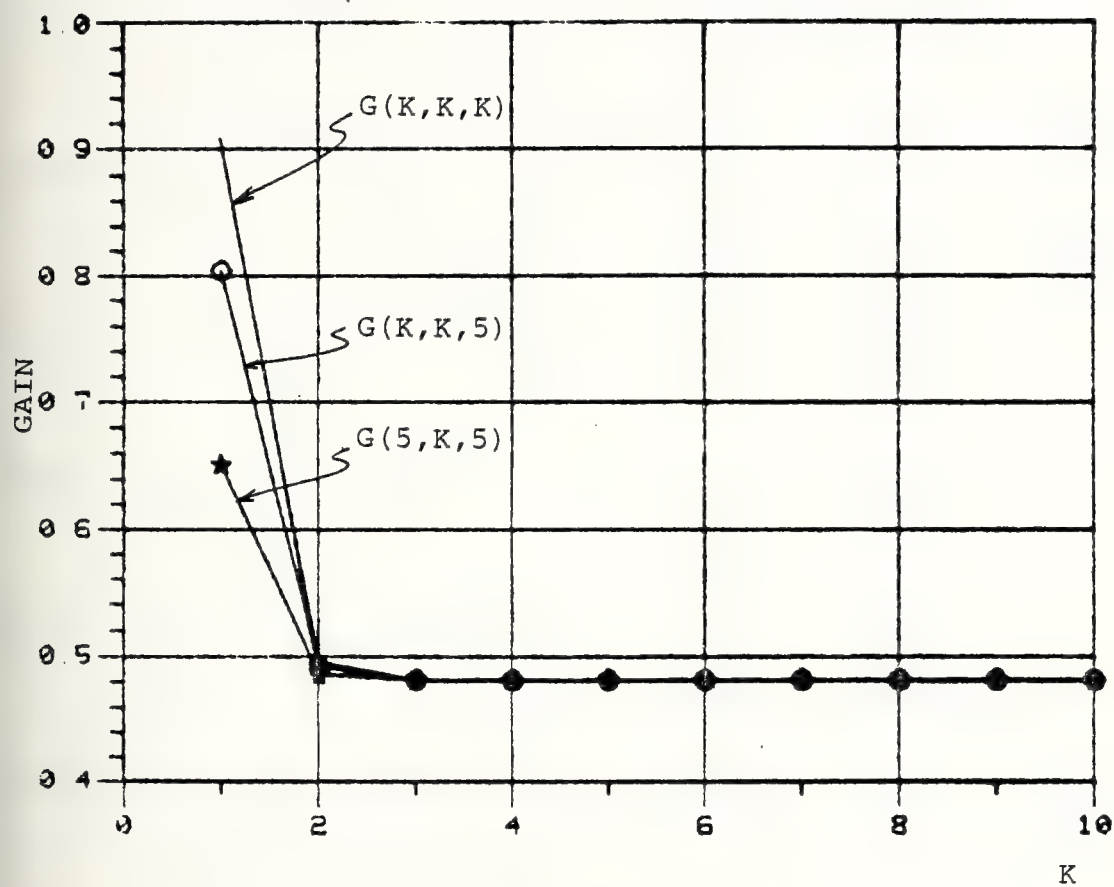


Figure 4.5

Gains for the Time-Frame

SIGNAL/NOISE = 30 dB

FRAME CORRELATION: $\rho_v = \rho_h = 0.96$

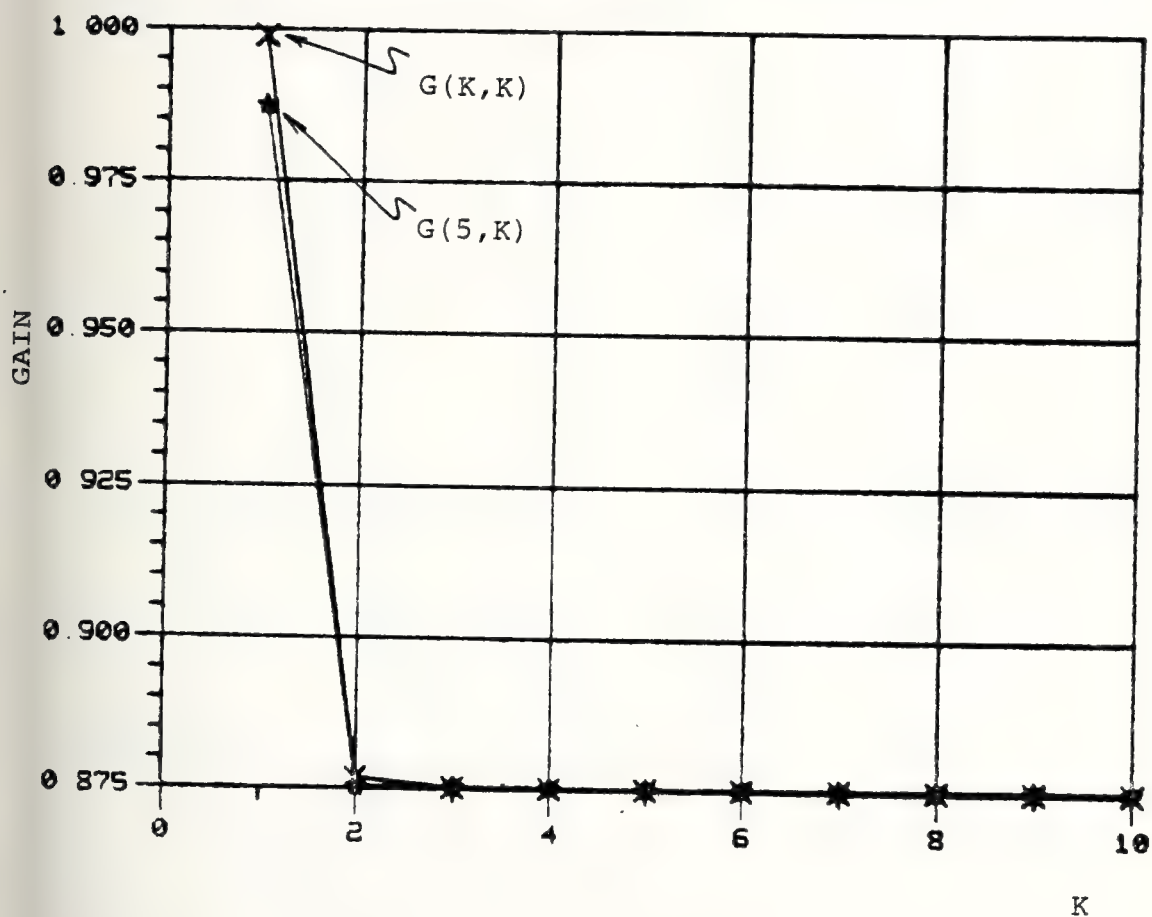


Figure 4.6
Gains for the First Frame

SIGNAL/NOISE = 30 dB

FRAMES CORRELATION: $\rho_v = \rho_h = \rho_t = 0.96$

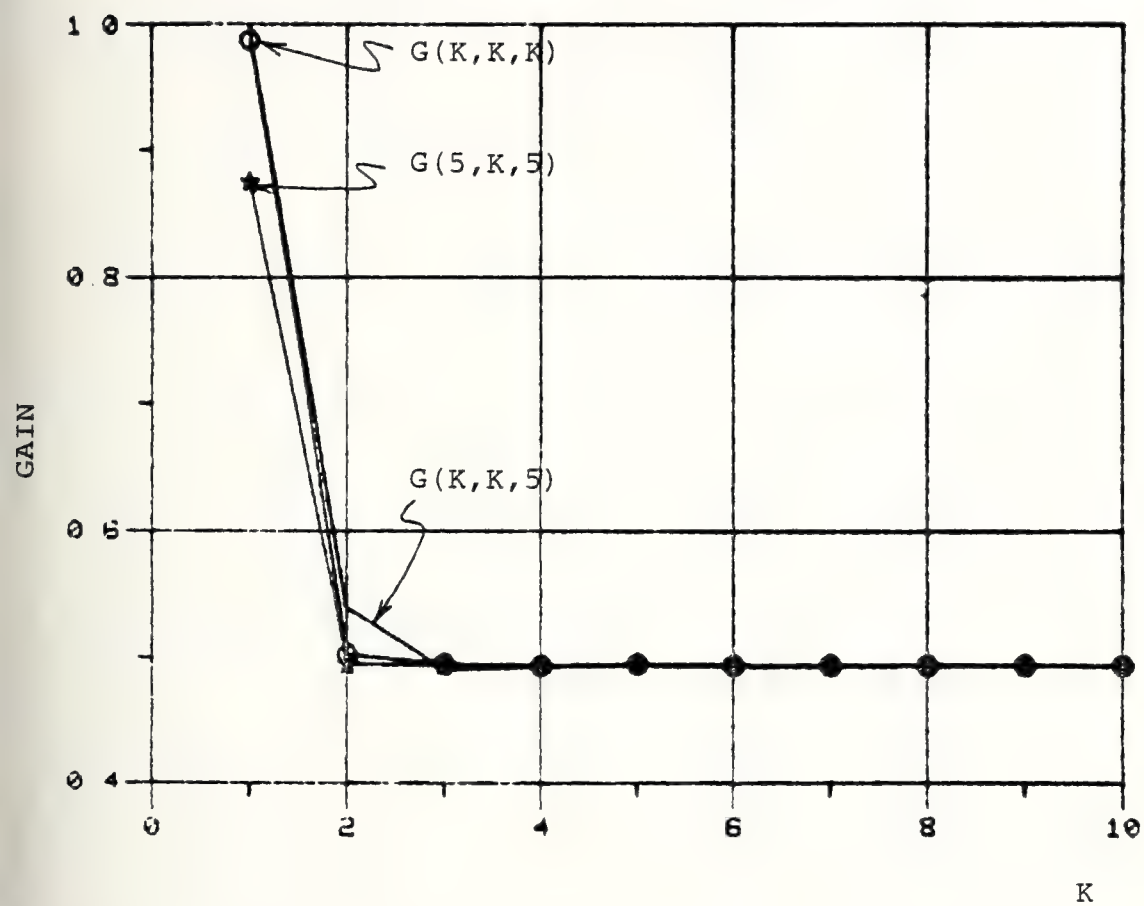


Figure 4.7

Gains for the Time-Frame

SIGNAL/NOISE = 3 dB

FRAMES CORRELATION: $\rho_v = \rho_h = 0.8$

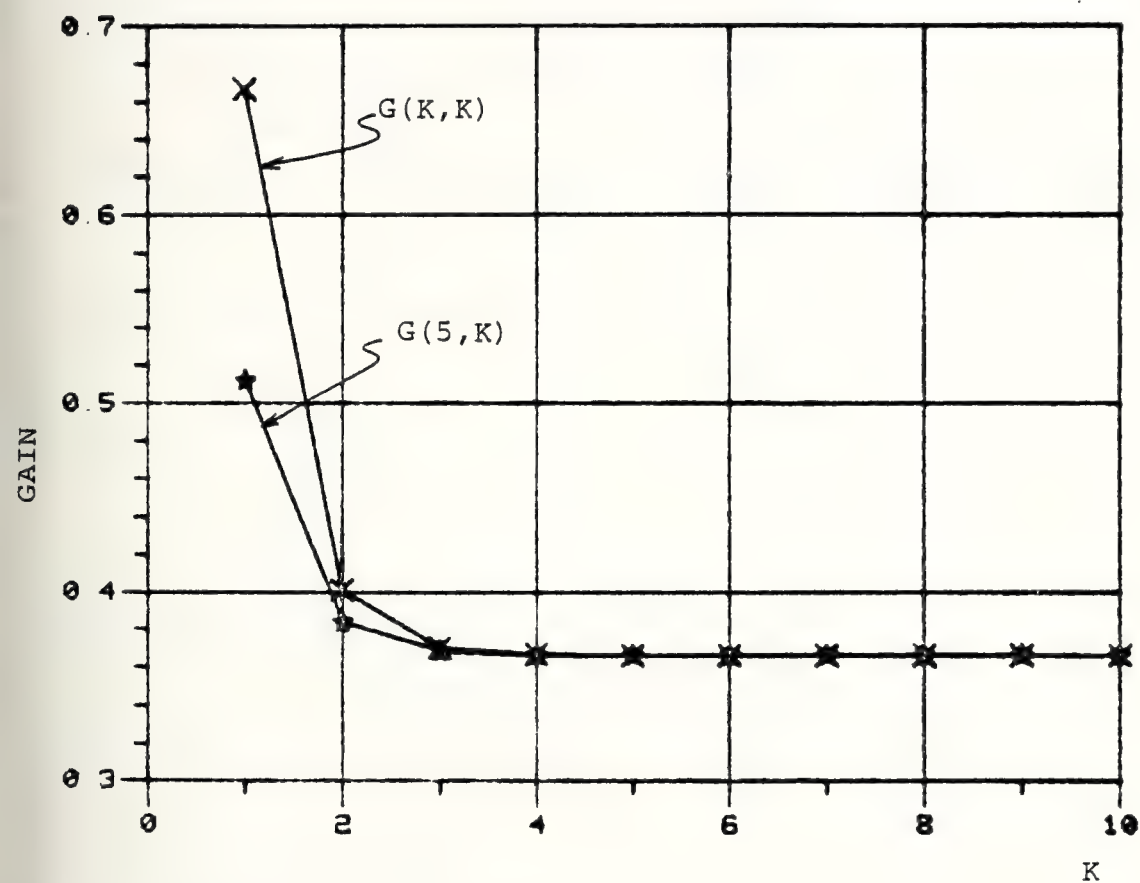


Figure 4.8

Gains for the First Frame

SIGNAL/NOISE = 3 dB

FRAMES CORRELATION: $\rho_v = \rho_h = \rho_t = 0.8$

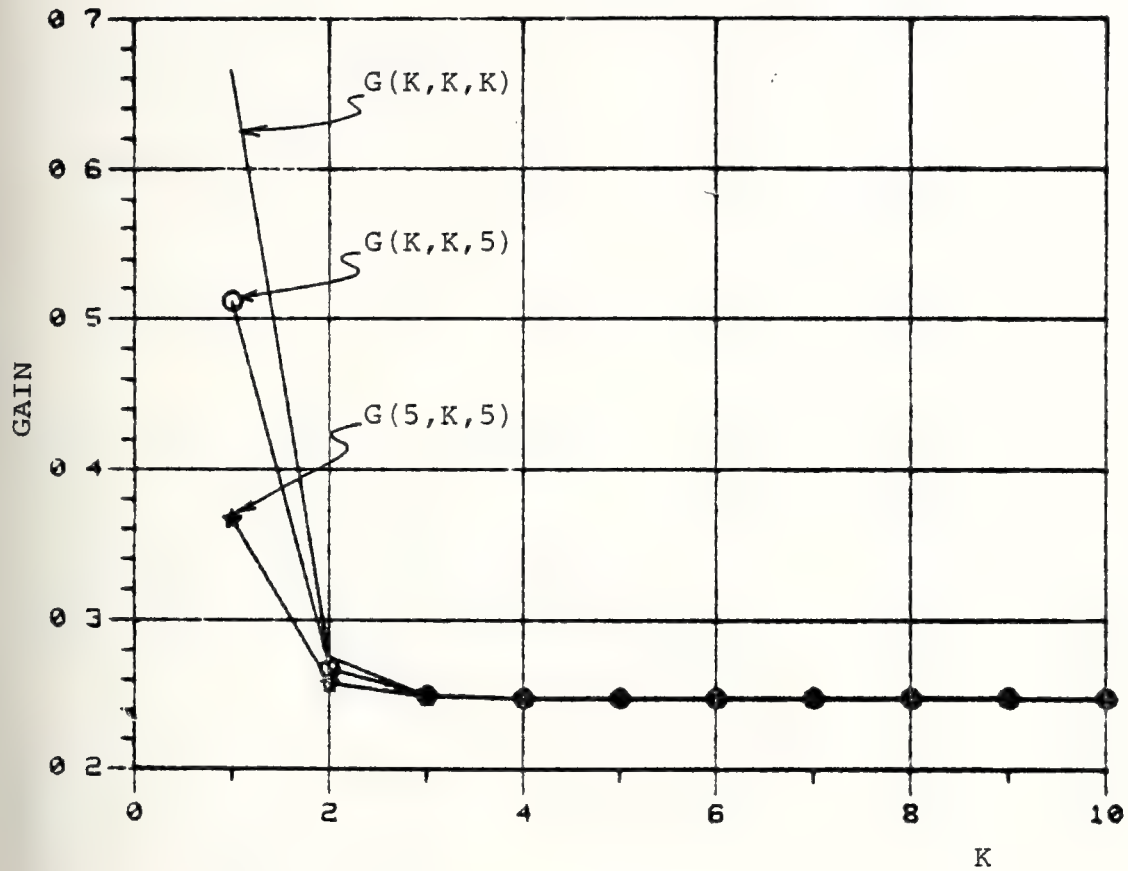


Figure 4.9

Gains for the Time-Frame

Table 4.1

| | Error Variance | | | |
|--|------------------|------------|----------------|------------|
| Noise Variance | Three-Dimensions | | Two-Dimensions | |
| | Estimation | Prediction | Estimation | Prediction |
| 1.0 | .236 | .293 | .357 | .554 |
| 0.8 | .210 | .267 | .318 | .527 |
| 0.6 | .178 | .236 | .271 | .493 |
| 0.4 | .140 | .197 | .212 | .450 |
| 0.2 | .089 | .144 | .132 | .387 |
| Picture Variance = 1.00 Spatial Correlation = 0.70 Time Correlation = 0.95 | | | | |

Table 4.2

| | Error Variance | | | |
|---|------------------|------------|----------------|------------|
| Noise Variance | Three-Dimensions | | Two-Dimensions | |
| | Estimation | Prediction | Estimation | Prediction |
| 1.0 | .269 | .351 | .301 | .427 |
| 0.8 | .240 | .324 | .268 | .400 |
| 0.6 | .206 | .293 | .229 | .367 |
| 0.4 | .162 | .251 | .180 | .324 |
| 0.2 | .103 | .191 | .114 | .263 |
| Picture Variance = 1.0 Spatial Correlation = 0.8 Time Correlation = 0.8 | | | | |

Table 4.3

| | Error Variance | | | |
|---|------------------|------------|----------------|------------|
| Noise Variance | Three-Dimensions | | Two-Dimensions | |
| | Estimation | Prediction | Estimation | Prediction |
| 1.0 | .280 | .376 | .301 | .427 |
| 0.8 | .250 | .349 | .268 | .400 |
| 0.6 | .214 | .317 | .229 | .367 |
| 0.4 | .168 | .274 | .180 | .324 |
| 0.2 | .107 | .213 | .114 | .263 |
| Picture Variance = 1.0 Spatial Correlation = 0.8 Time Correlation = 0.7 | | | | |

V. HYBRID FILTERS

A. INTRODUCTION

In this chapter a new class of image filters is introduced, called hybrid filters. The recursive filters, presented in Chapters III and IV, utilize only part of the data set, arbitrarily defined as "past" measurements. Since the whole picture is often available, an optimal smoother might be sought in order to utilize all the information available. For the two-dimensional recursive filters of Chapter III, the smoother would be the optimum combination of the four estimates obtained by scanning the picture starting at each corner. The first difficulty is that those filters are not optimum resulting in a sub-optimum smoother. Second, the smoother would require scanning the picture four times, therefore increasing substantially its complexity of implementation.

The hybrid filter introduced here is a smoother that combines optimally the estimate of the recursive filter (two or three dimensions) with an arbitrary set of "future" measurements. It will be applied in picture enhancement and compared against the recursive filter as well as the non-recursive filter presented in [6]. Some examples of hybrid filters designed to predict the pixel gray level will also be presented. These filters will be applied in Chapter VI for purposes of target detection.

The hybrid filter is particularly useful in some applications where the pictures are not from a fixed scenario. For example, the camera has a movement in order to follow some target. In such cases, the background may change so much that past frames don't carry enough information about the present frame. In this case, the three-dimensional recursive filter can't be applied, but a hybrid filter can be designed using recursively the measurements in the present frame, and, non-recursively, the "crude" measurements in the previous frame.

B. TWO-DIMENSIONAL HYBRID FILTERS

In this section we present two examples of hybrid filters in two-dimensions. One is a smoother designed to enhance pictures contaminated with additive white noise. The other is a filter designed to predict the pixel gray level, based on noisy measurements of previously scanned pixels. Here we assume the picture is scanned row by row from top to bottom.

1. Two-Dimensional Hybrid Smoother

The recursive filters presented in Chapter III utilize the measurements in region Y_{mn} (see figure 3.2) to have an estimate $\hat{X}(m,n)$. To improve this estimate we are going to use also a selected set of measurements in the neighborhood of (m,n) , say \underline{Y} , and combine optimally with $\hat{X}(m,n)$, in order to have a better estimate $\hat{X}_1(m,n)$. Let's choose for \underline{Y} the set of five neighbors shown in figure 5.1.

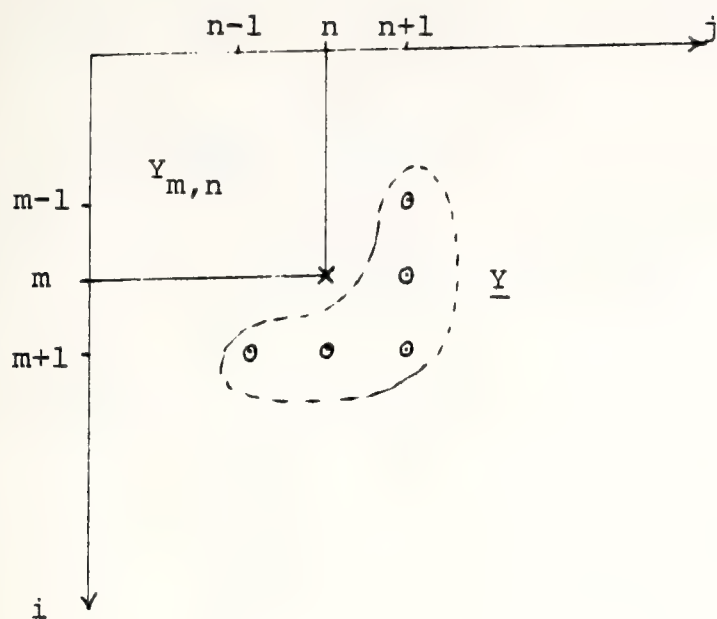


Figure 5.1
Two-dimensional hybrid smoother

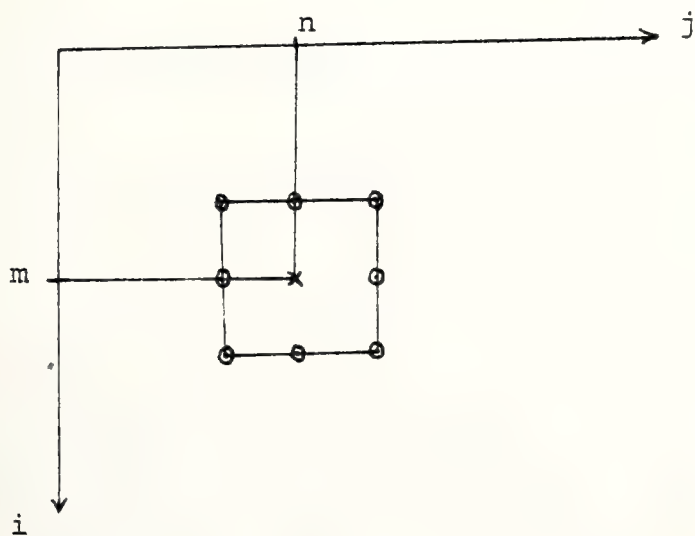


Figure 5.2
Non-recursive filter

Thus

$$\hat{X}_1(m,n) = \underline{h}^T \underline{g} \quad (5.1)$$

where

$$\underline{g} = \begin{bmatrix} \hat{X}(m,n) \\ y(m+1,n-1) \\ y(m+1,n) \\ y(m+1,n+1) \\ y(m,n+1) \\ y(m-1,n+1) \end{bmatrix} \quad \underline{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix}$$

Let's calculate the weights \underline{h} in order to minimize the mean-square-error in the estimate. The error is

$$e(m,n) = X - \hat{X}_1 = X - \underline{h}^T \underline{g}$$

The error variance is

$$P(m,n) = E[(X - \underline{h}^T \underline{g})^2] \quad (5.2)$$

Differentiating (5.2) with respect to \underline{h} and equating to zero

$$\underline{h} = R_{gg}^{-1} R_{Xg} \quad (5.3)$$

where

$$R_{gg} = E(qq^T)$$

$$R_{Xg} = E(Xg)$$

The measurement equation is

$$y(m,n) = X(m,n) + v(m,n) \quad (5.4)$$

where v is zero-mean white noise, uncorrelated with X , having variance

$$E[v^2(m,n)] = r$$

The gray level X is a zero-mean (or known mean) homogeneous random field with autocorrelation

$$R(i,j) = E[X(k,\ell)X(k+i,\ell+j)] \quad (5.5)$$

The elements of the autocorrelation matrix R_{gg} , that don't include $\hat{X}(m,n)$, are easily found using equations (5.4) and (5.5). To find the elements which include \hat{X} , observe that in Chapter III a method was developed to calculate error variance of recursive filters [3,5], where $E(\hat{X}^2)$ is also computed as well as $E(X\hat{X})$. It can be verified that the same method can be used for higher order

recursive filters, though in this derivation we restrict ourselves to recursive filter [3].

Therefore, using (3.11) and (3.13):

$$\begin{aligned} E(\hat{X}^2) &= \sigma_e^2 \\ E(\hat{X}\hat{X}) &= b \end{aligned} \tag{5.6}$$

It can be verified that the covariances between \hat{X} and the measurements \underline{y} can be expressed as a function of b given by

$$\begin{aligned} E[\hat{X}y(m+1, n-1)] &= \rho_v \rho_h b \\ E[\hat{X}y(m+1, n)] &= \rho_v b \\ E[\hat{X}y(m+1, n+1)] &= \rho_v \rho_h b \\ E[\hat{X}y(m, n+1)] &= \rho_h b \\ E[\hat{X}y(m-1, n-1)] &= \rho_v \rho_h b \end{aligned} \tag{5.7}$$

Therefore we have

$$R_{gg} = \begin{bmatrix} \sigma_e^2 & \rho_v \rho_h b & \rho_v b & \rho_v \rho_h b & \rho_h b & \rho_v \rho_h b \\ & \sigma^2 + r & \rho_h \sigma^2 & \rho_h^2 \sigma^2 & \rho_v \rho_h^2 \sigma^2 & \rho_v^2 \rho_h^2 \sigma^2 \\ & & \sigma^2 + r & \rho_h \sigma^2 & \rho_h \rho_v \sigma^2 & \rho_v^2 \rho_h \sigma^2 \\ & & & \sigma^2 + r & \rho_v \sigma^2 & \rho_v^2 \sigma^2 \\ & & & & \sigma^2 + r & \rho_v \sigma^2 \\ & & & & & \sigma^2 + r \end{bmatrix} \quad (5.8)$$

$$R_{Xg} = \begin{bmatrix} b \\ \rho_v \rho_h \sigma^2 \\ \rho_v \sigma^2 \\ \rho_v \rho_h \sigma^2 \\ \rho_h \sigma^2 \\ \rho_v \rho_h \sigma^2 \end{bmatrix}$$

where we have made use of

$$R(i,j) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|}$$

a. Performance and Comparison

The error variance of the hybrid smoother is calculated substituting (5.3) into (5.2):

$$P(m,n) = \sigma^2 - \underline{h}^T R_{Xg} \quad (5.9)$$

At this point it is interesting to compare the hybrid smoother and the recursive filter [3], since we know how to calculate the error variance for both. It is also interesting to include in such comparison the non-recursive filter presented in [6]. This filter uses a small window of 3×3 , shown in figure 5.2, to estimate the pixel gray level at the center, minimizing the mean-square-error. The hybrid smoother must have superior performance, since it utilizes the same measurements of [3] and [6] together. However, we can't say which is better whether [3] or [6], since they use distinct data sets. Although filter [3] uses a larger data set than [6], the former uses all the closest adjacents to the estimated pixel (m,n) .

In table 5.1 some numerical results are shown to help the comparison. As expected, the hybrid filter presented the best performance with error variance about 25% smaller (for $\rho = 0.94$) than filter [6]. The non-recursive filter [6] presented better performance than the recursive filter [3]. The error variance of the optimum interpolator constrained to the observations in $Y_{m,n}$ is also included in

Table 5.1

| | Error Variance | | | |
|------------------------|----------------|------------|------------|----------------------|
| Correlation | Hybrid | Filter [6] | Filter [3] | Optimum Interpolator |
| 0.60 | 0.334 | 0.346 | 0.400 | 0.399 |
| 0.70 | 0.279 | 0.298 | 0.357 | 0.355 |
| 0.80 | 0.217 | 0.245 | 0.300 | 0.294 |
| 0.90 | 0.145 | 0.182 | 0.218 | 0.205 |
| 0.92 | 0.128 | 0.167 | 0.197 | 0.181 |
| 0.94 | 0.112 | 0.152 | 0.173 | 0.155 |
| 0.96 | 0.111 | 0.136 | 0.146 | 0.124 |
| Picture Variance = 1.0 | | | | |
| Noise Variance = 1.0 | | | | |

this table. It is interesting to see that the use of just 5 non-causal observations by the hybrid filter were enough to fully compensate the suboptimality of the recursive filter [3]. Observe that the 9-point non-recursive filter [6] also presented better performance than the optimum interpolator, even for spatial correlation as high as 0.94. This indicates that most of the information about $X(m,n)$ resides in its nearest neighbors. This motivates the derivation of simple filters like the hybrid filter introduced here. It is much simpler than the optimum recursive filter [9] with insignificant loss in performance. It also has

the same order of complexity as the non-recursive filter [6], but superior performance. Complexity here means the number of multiplications, which is 9 for the non-recursive filter and 10 for the hybrid filter.

The hybrid filter may be seen as a way of artificially increasing the window of filter [6].

b. Experimental Results

In this section we present the results of some experiments where the filters were used in picture enhancement. In this experiment, the picture of figure 5.3 was contaminated with white noise. Assuming knowledge of the noise variance, the picture was modeled by the method presented in Chapter II. With this model and noise variance, we designed the three filters: hybrid, recursive and non-recursive. The theoretical performance was computed for each filter, as in the last item. A signal-to-noise ratio (S/N) was defined as the ratio between the variance of the original picture (figure 5.3) and the variance of the noise. A processing gain was defined as the ratio of (S/N)'s at the filters output and input:

$$F = \frac{(S/N)_o}{(S/N)_i}$$

Table 5.2 summarizes the results. It can be seen that the computed performance is pretty close to the

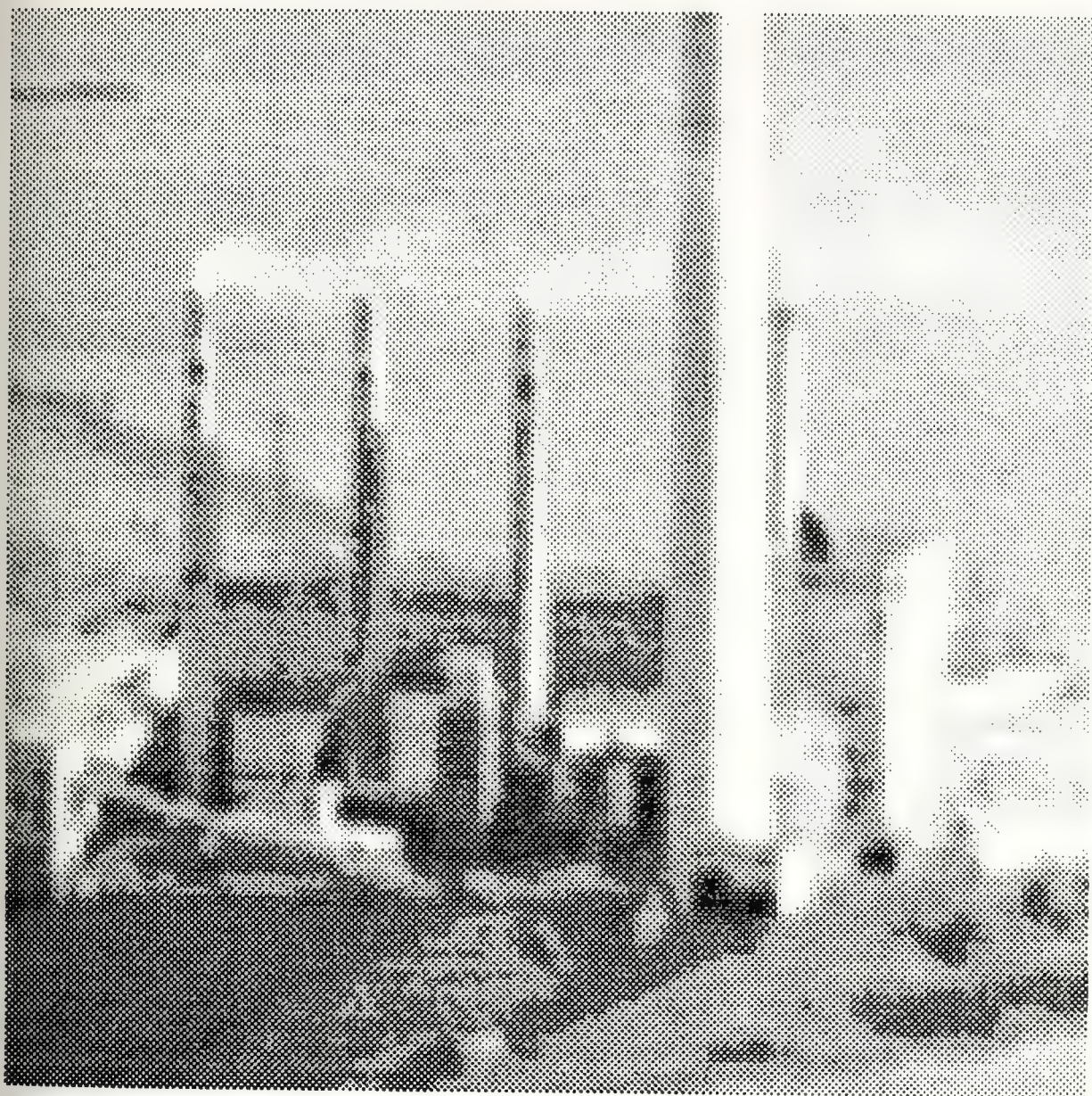


Figure 5.3
Original image of a power plant

Table 5.2

| (S/N) _i dB | HYBRID | | | NON-RECURSIVE | | | | RECURSIVE | | | |
|--------------------------|-----------------------|-------|--------|-----------------------|-------|--------|--|-----------------------|-------|--------|--------|
| | (S/N) _o dB | | F (dB) | (S/N) _o dB | | F (dB) | | (S/N) _o dB | | Theory | F (dB) |
| | Theory | Meas. | | Theory | Meas. | | | Theory | Meas. | | |
| +3 | 10.8 | 11.5 | 8.5 | 9.7 | 10.4 | 7.4 | | 8.8 | 9.3 | 6.3 | |
| 0 | 9.1 | 9.8 | 9.8 | 7.9 | 8.2 | 8.2 | | 7.2 | 7.8 | 7.8 | |
| -3 | 7.4 | 8.1 | 11.1 | 6.0 | 5.8 | 8.8 | | 5.8 | 6.6 | 9.6 | |

(S/N)_i - signal-to-noise ratio before the filter

(S/N)_o - signal-to-noise ratio after the filter

$$F = \frac{(S/N)_o}{(S/N)_i} - \text{processing gain}$$

results of the experiments. The hybrid is clearly superior to the others and the non-recursive filter is better than the recursive, except for the very low S/N of -3 dB, when the former was a little better. However the subjective evaluation of the pictures (figures 5.4 through 5.15) is more difficult. In our evaluation the hybrid filter had superior performance, therefore validating the theoretical results, as well as the measurements of table 5.2. However, the results of the non-recursive filter, in our subjective evaluation, were inferior to the recursive filter in all three experiments, contradicting the measurements of table 5.2. Probably the mean-square-error is not a good figure of merit for visual perception, however there is no doubt that considerable improvement was achieved by processing the noisy pictures using such criteria.

2. Two-dimensional Hybrid Predictor

Assuming that the picture is scanned row by row, we may define "past" measurements as those pixels already scanned. Then the gray level of pixel (m,n) may be predicted using "past" measurements. In figure 5.16 the configuration of a hybrid filter designed to predict $X(m,n)$ is shown. This filter combines optimally the prediction $\hat{X}_p(m,n)$ of the recursive filter and the measurements \underline{Y} , shown in figure 5.16. The result is an improved prediction $\hat{X}(m,n)$ that utilizes all the information contained in the measurements Z_{mn} and \underline{Y} .

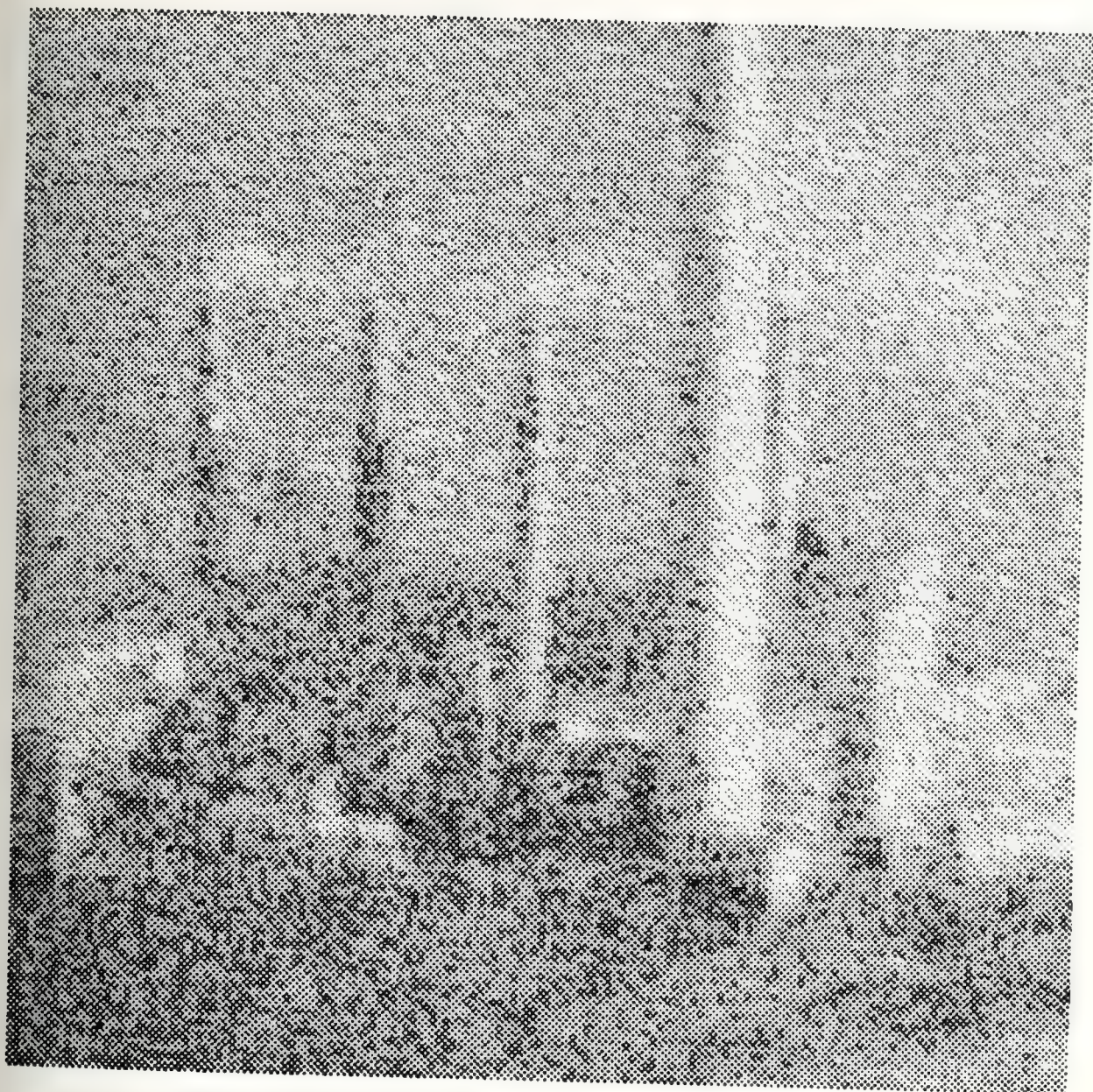


Figure 5.4

Image of figure 5.3 with additive noise (SNR = 3 dB)

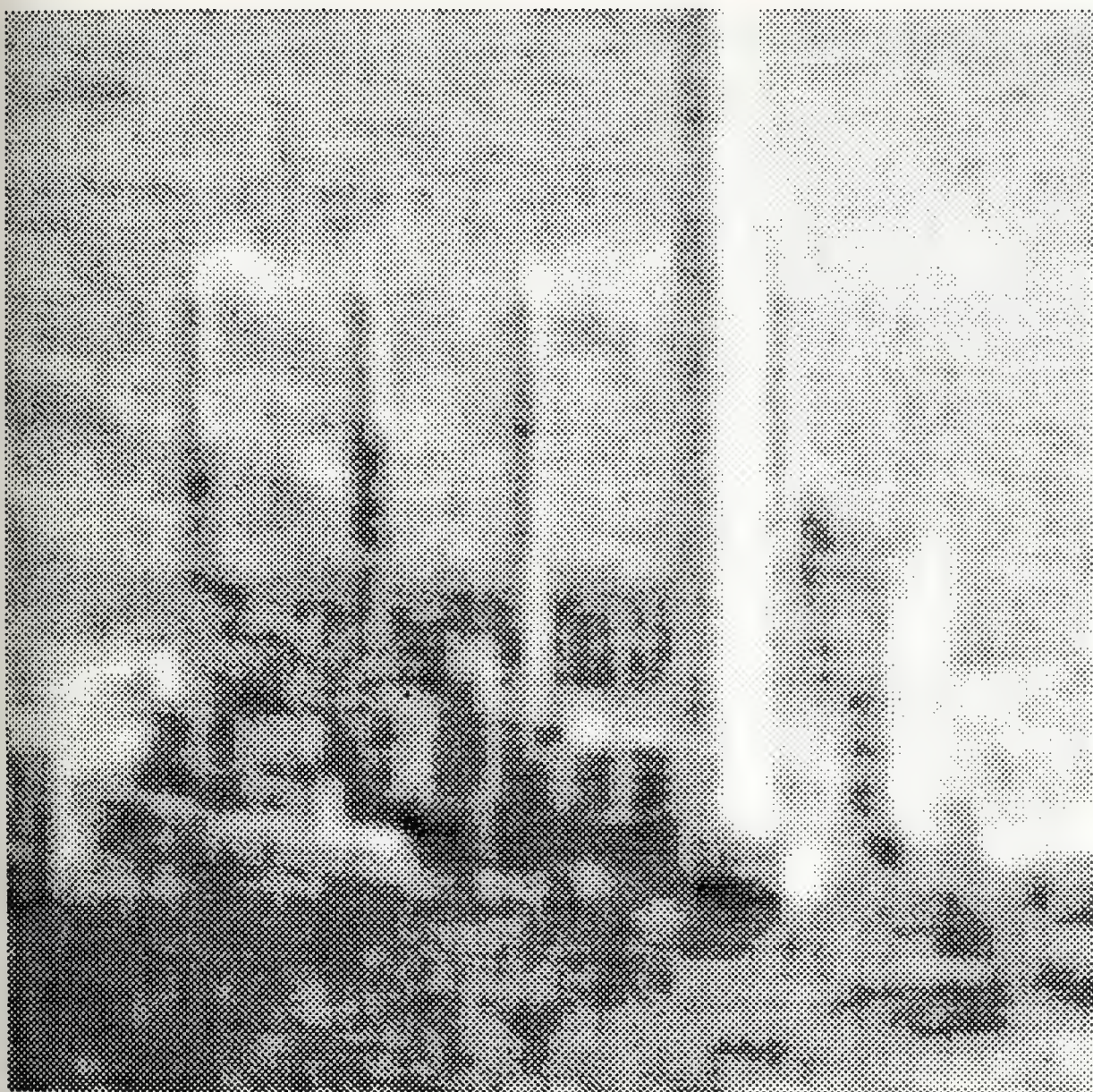


Figure 5.5

Image of figure 5.4 filtered by the hybrid filter
(SNR = 11.5 dB)

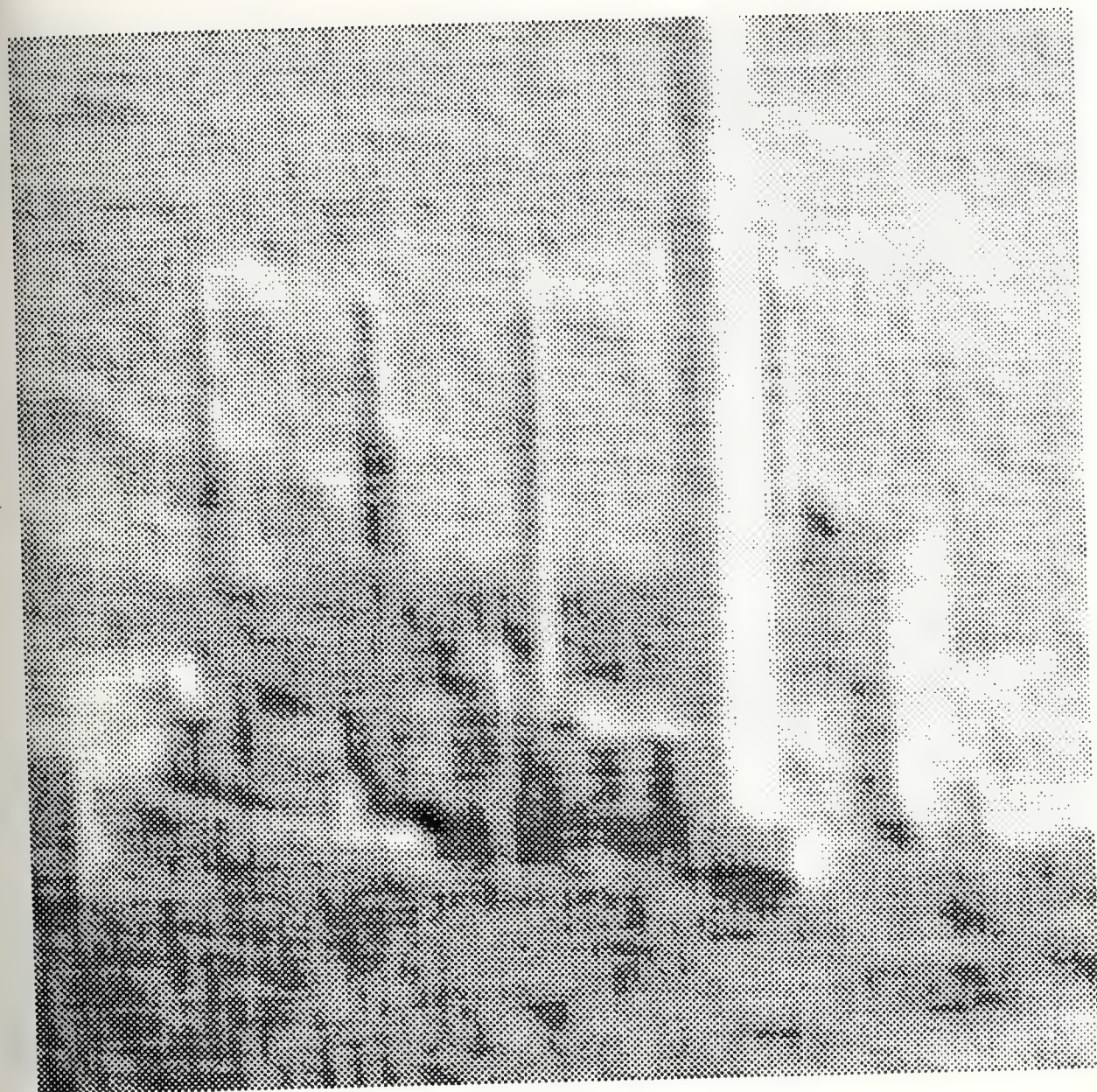


Figure 5.6

Image of figure 5.4 filtered by the recursive filter
(SNR = 9.3 dB)

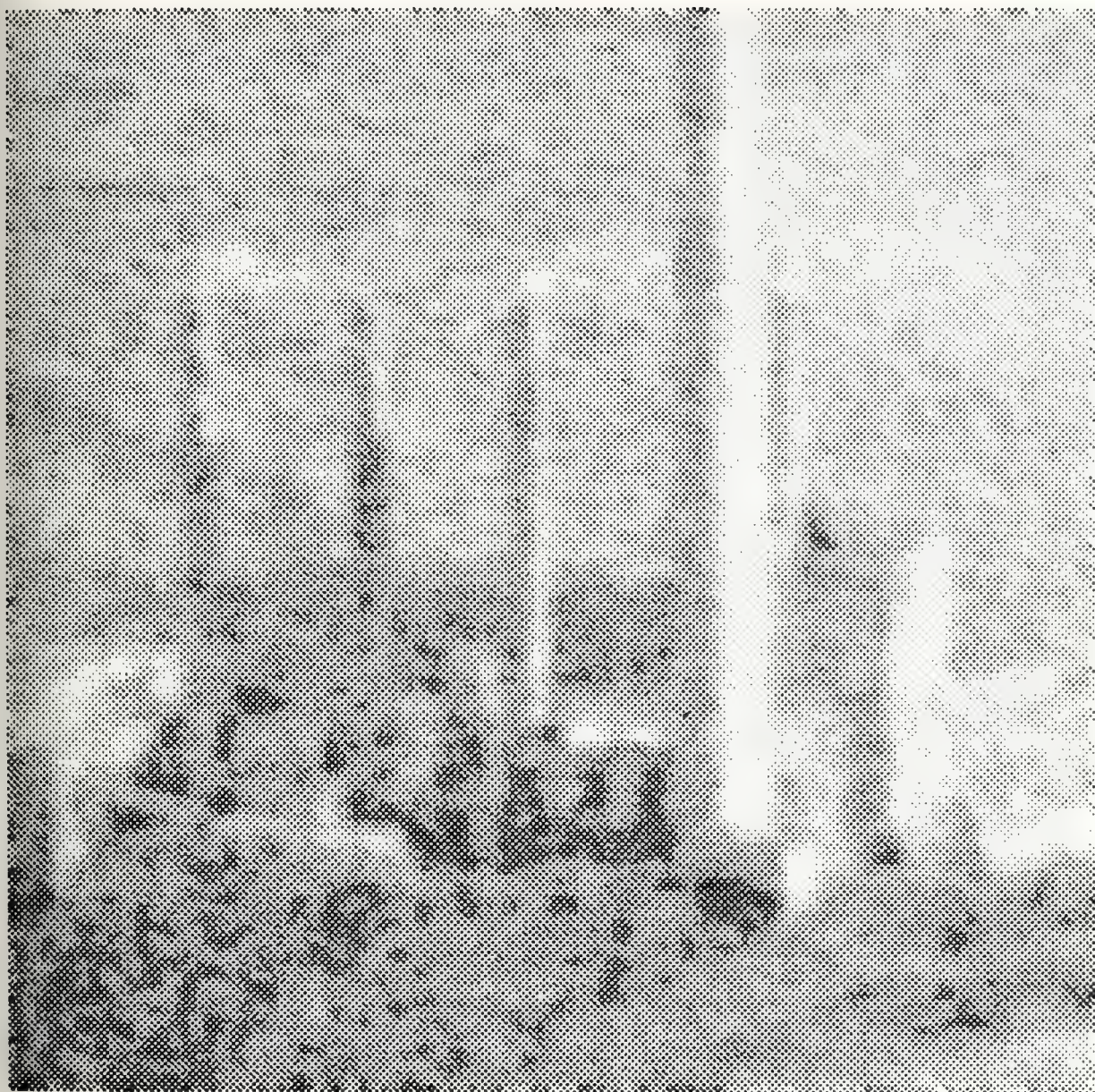


Figure 5.7

Image of figure 5.4 filtered by the 9-point non-recursive filter
(SNR = 10.4 dB)

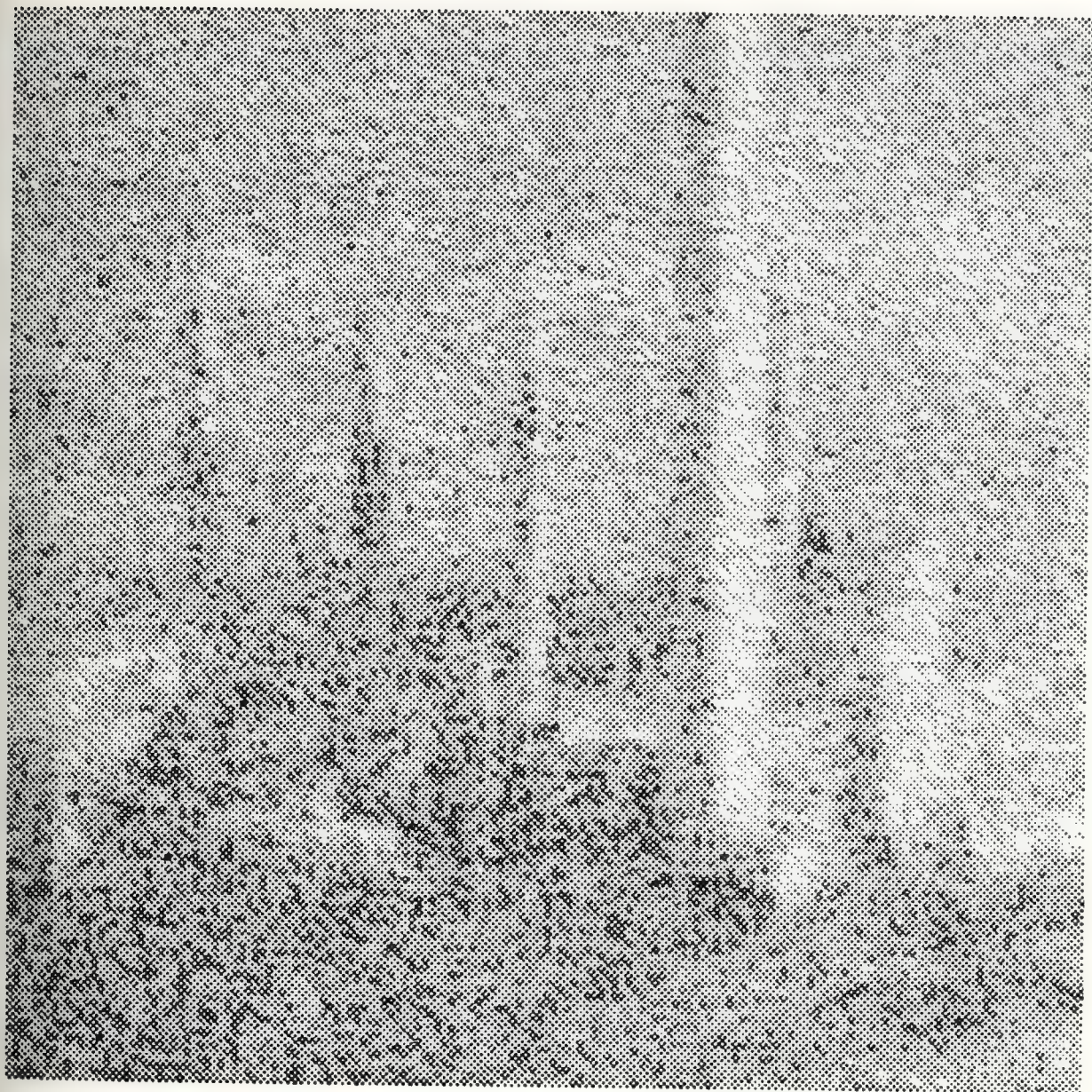


Figure 5.8

Image of figure 5.3 with additive noise (SNR = 0 dB)

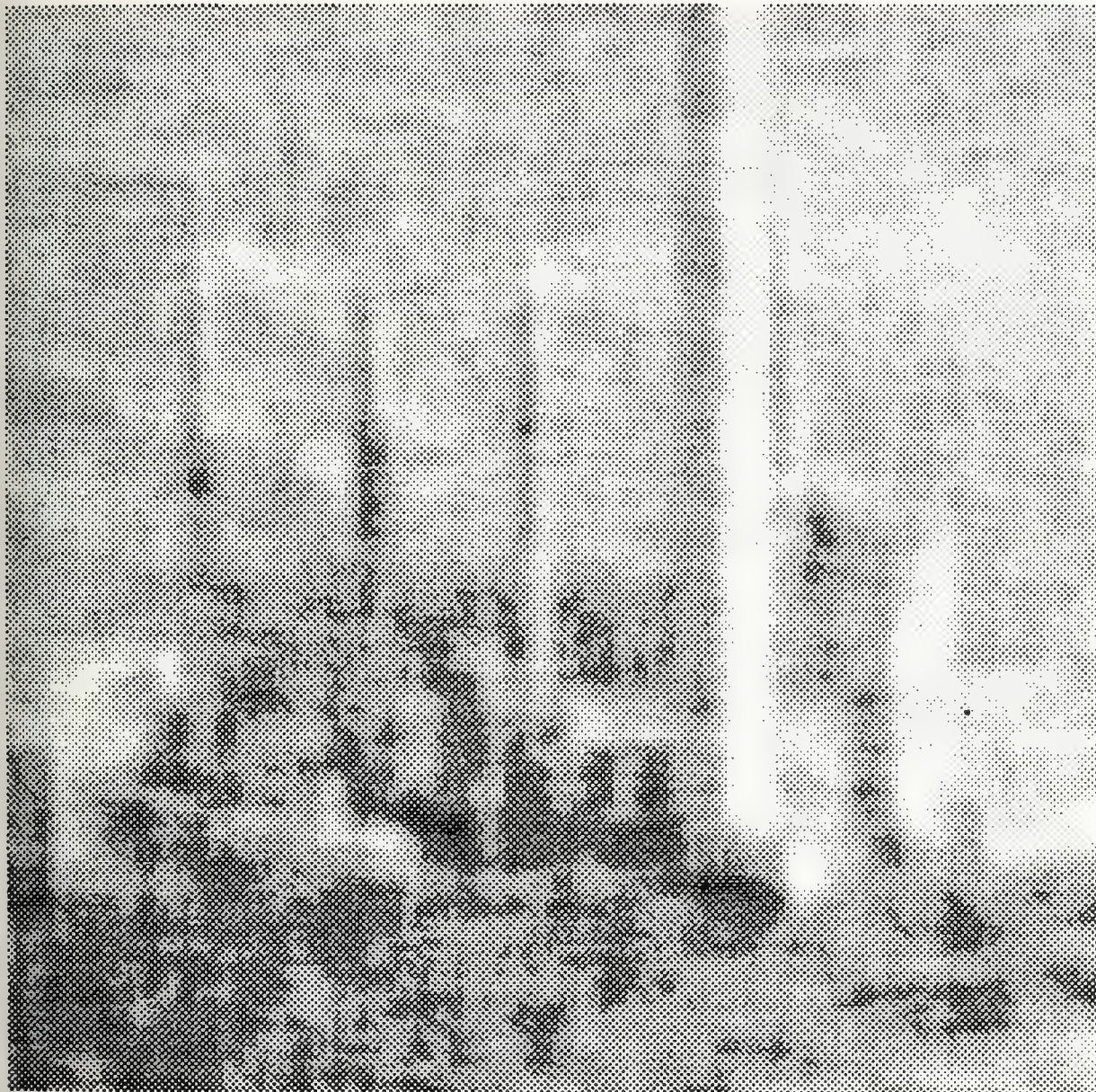


Figure 5.9

Image of figure 5.8 filtered by the hybrid filter

(SNR = 9.8 dB)

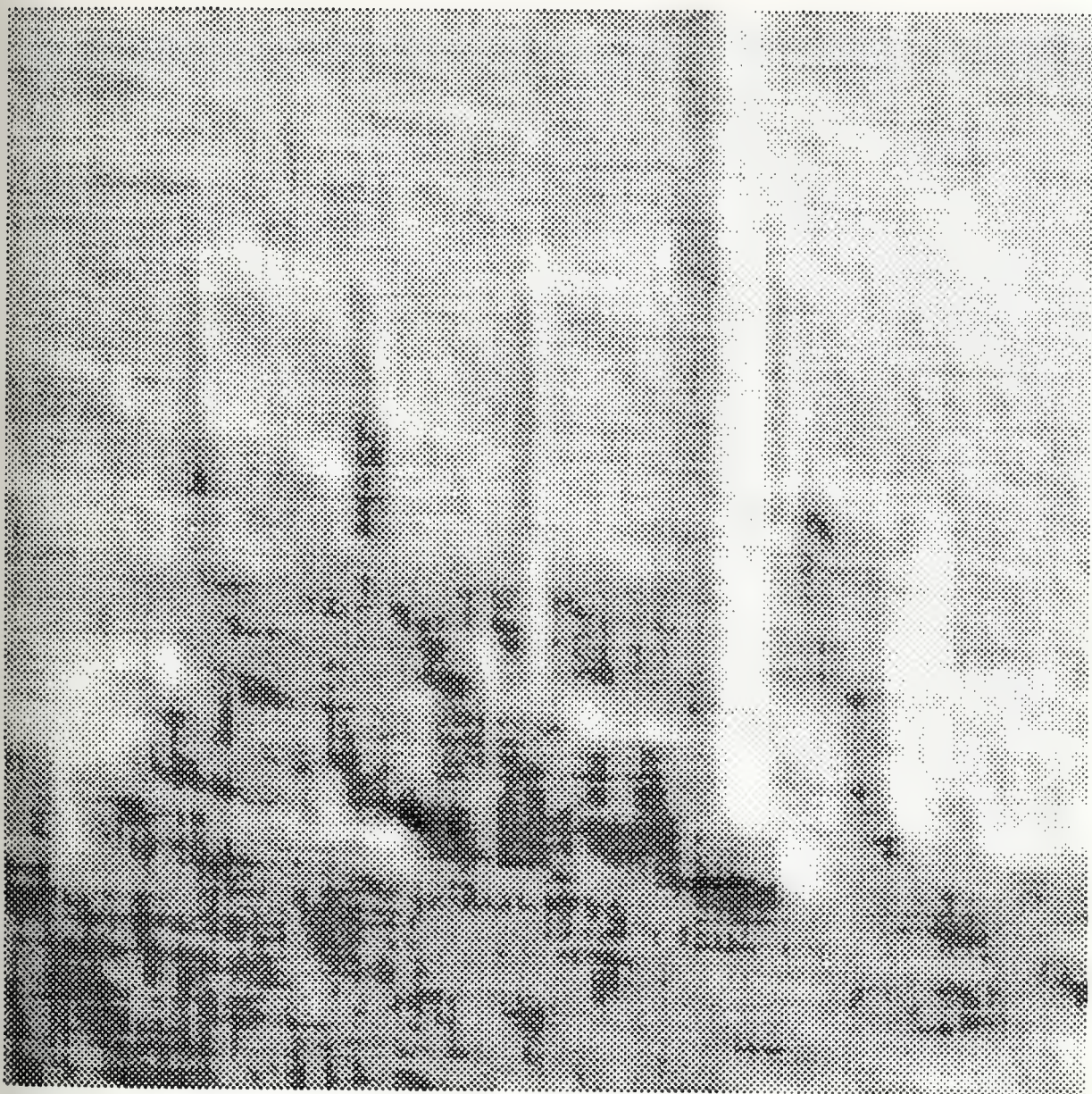


Figure 5.10

Image of figure 5.8 filtered by the recursive filter

(SNR = 7.8 dB)

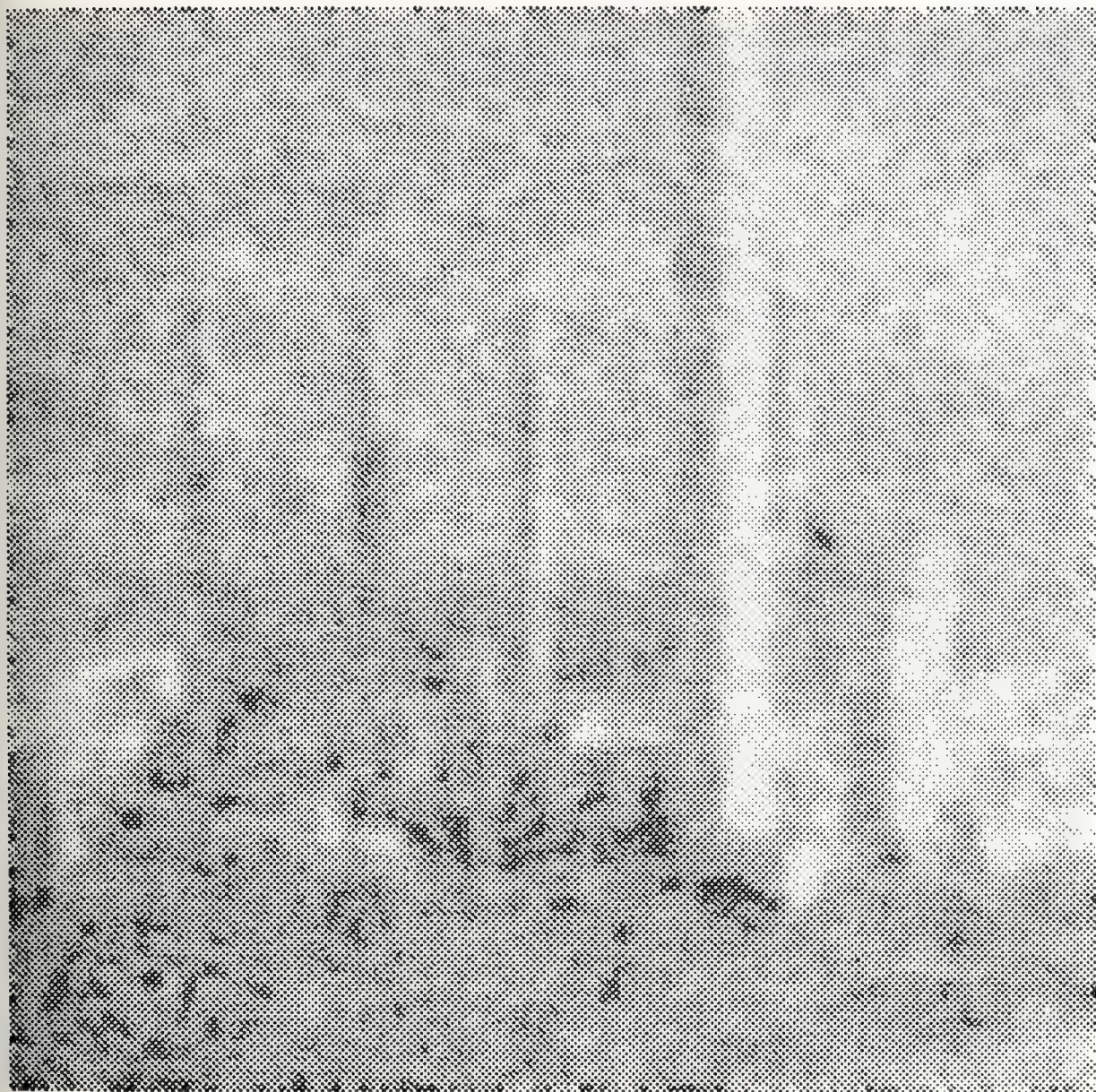


Figure 5.11

Image of figure 5.8 filtered by the 9-point non-recursive filter
(SNR = 8.2 dB)

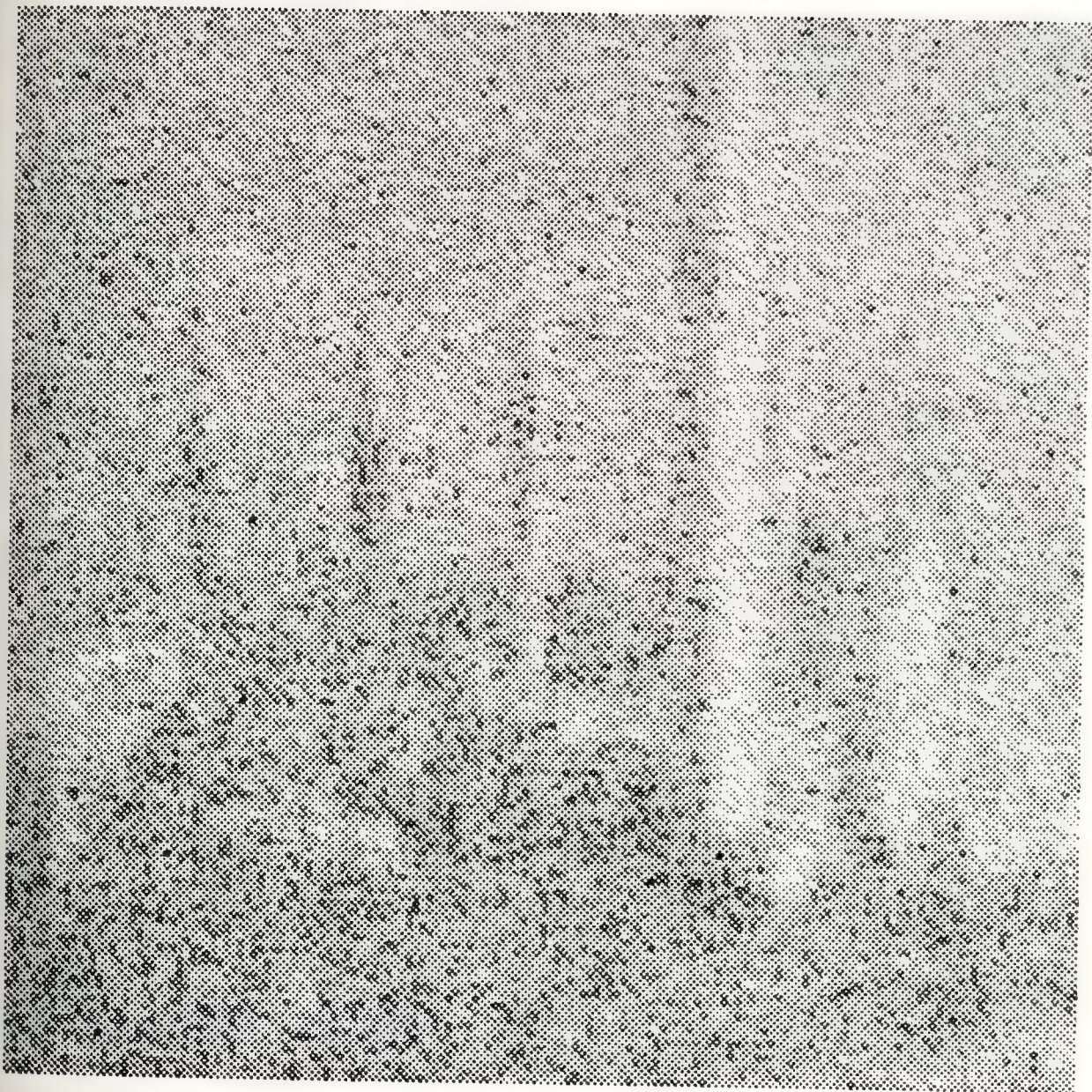


Figure 5.12

Image of figure 5.3 with additive noise (SNR = -3 dB)

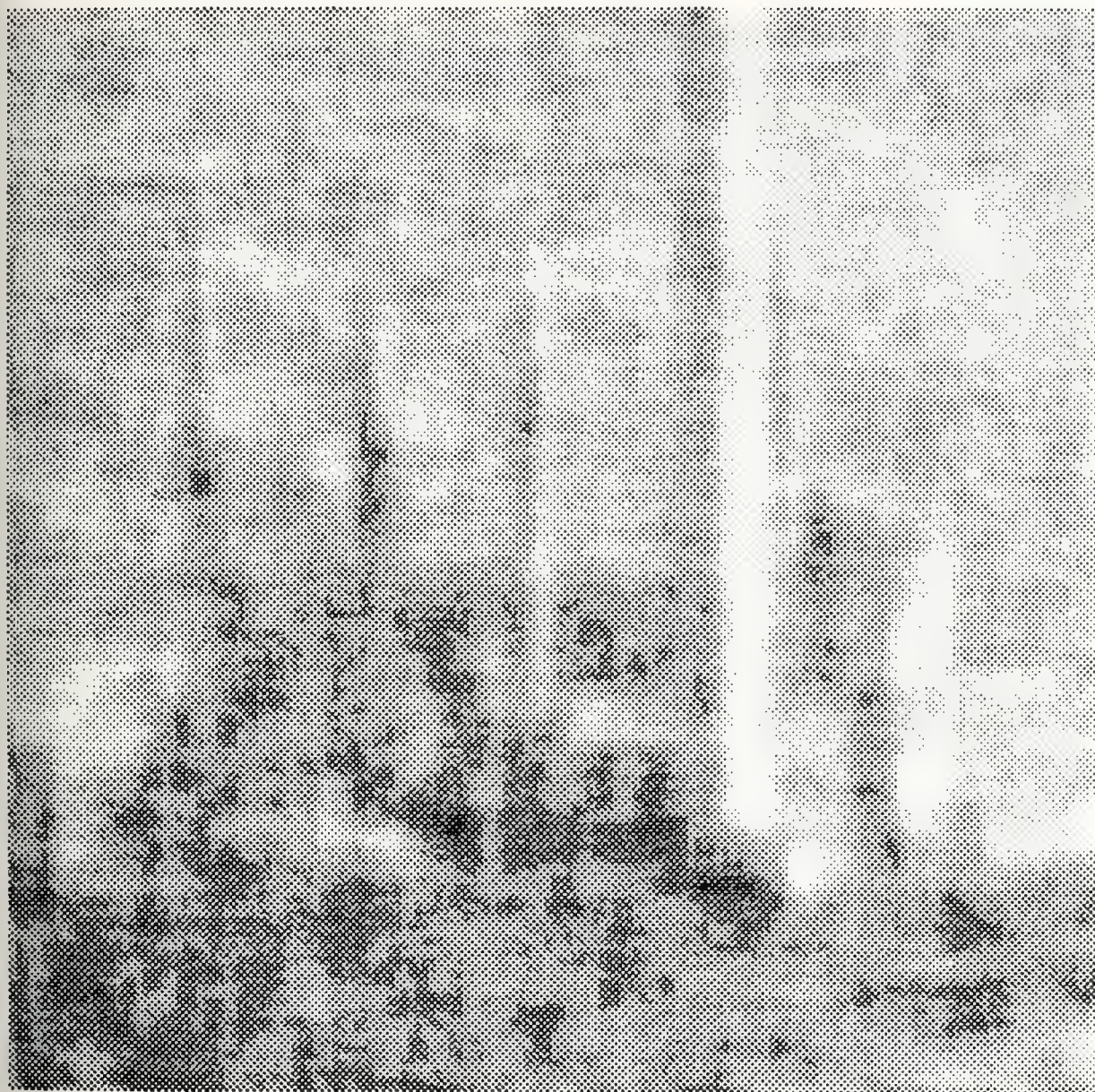


Figure 5.13

Image of figure 5.12 filtered by the hybrid filter
(SNR = 8.1 dB)

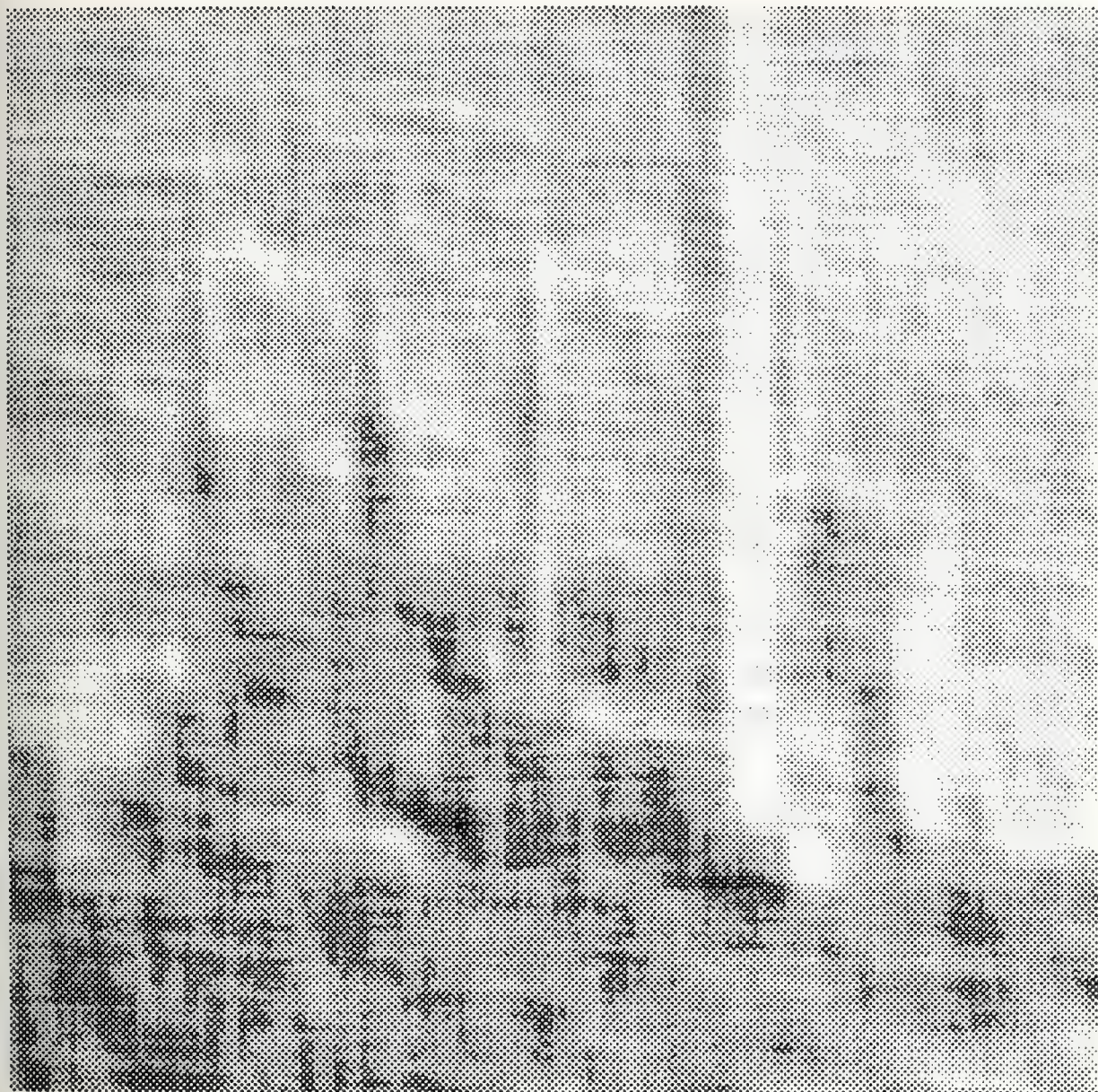


Figure 5.14

Image of figure 5.12 filtered by the recursive filter
(SNR = 6.6 dB)

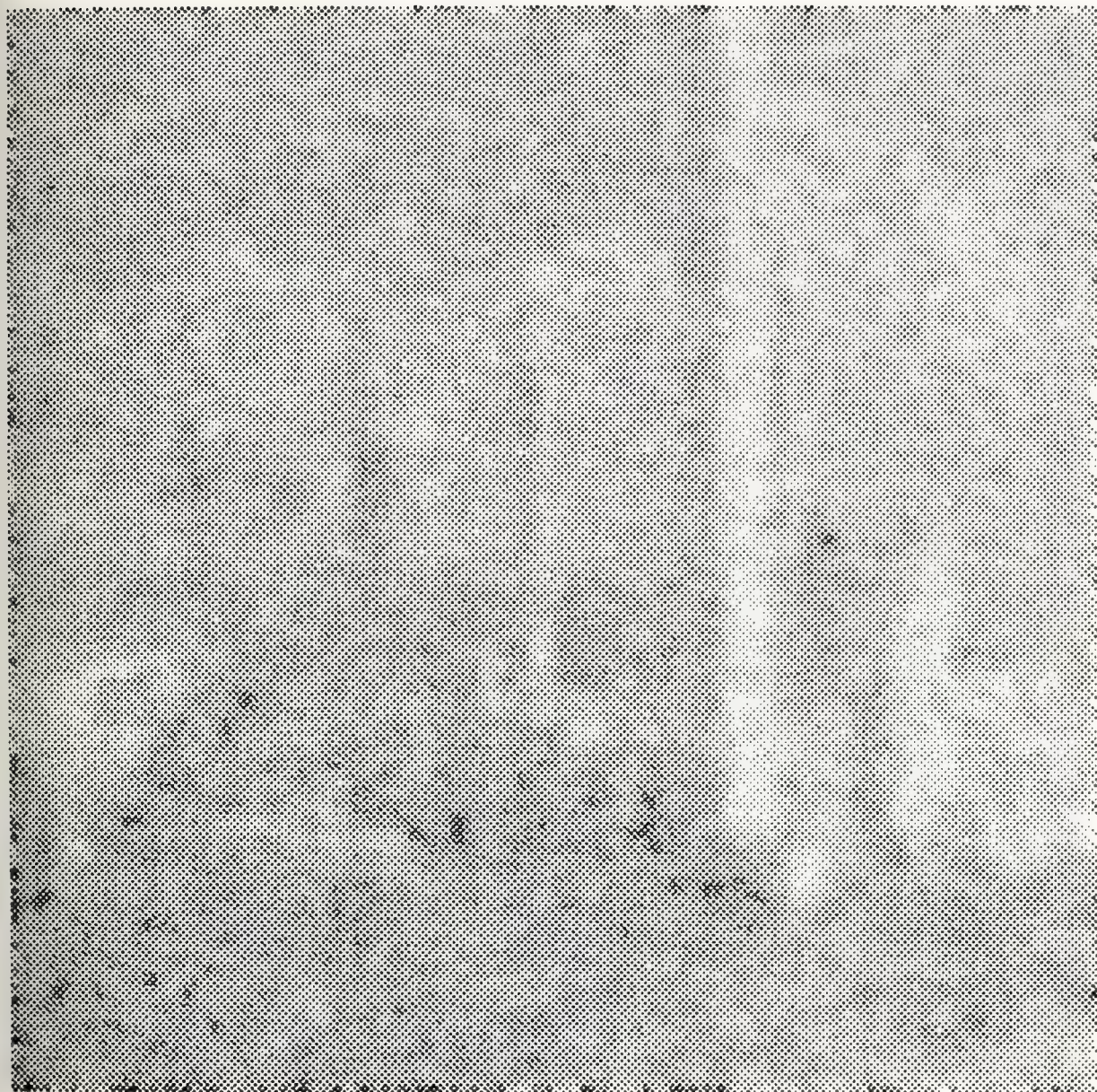


Figure 5.15

Image of figure 5.12 filtered by the 9-point non-recursive filter
(SNR = 5.8 dB)

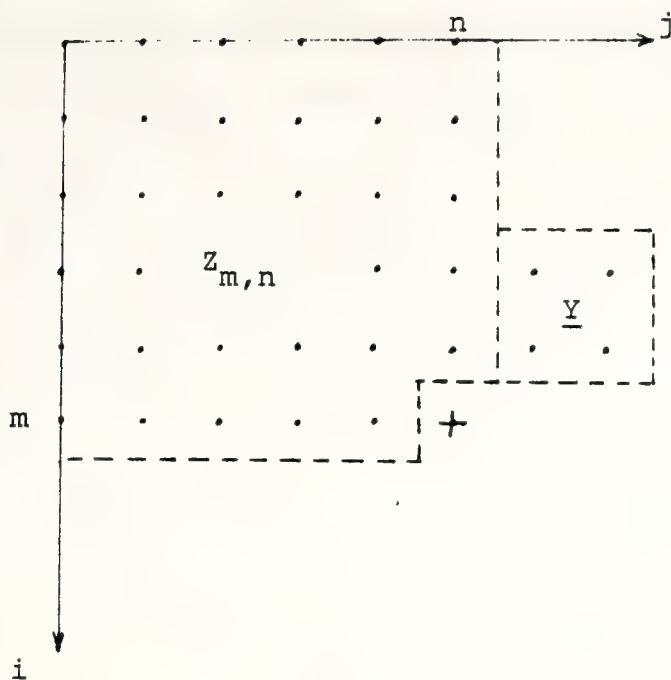


Figure 5.16

Two-dimensional hybrid predictor

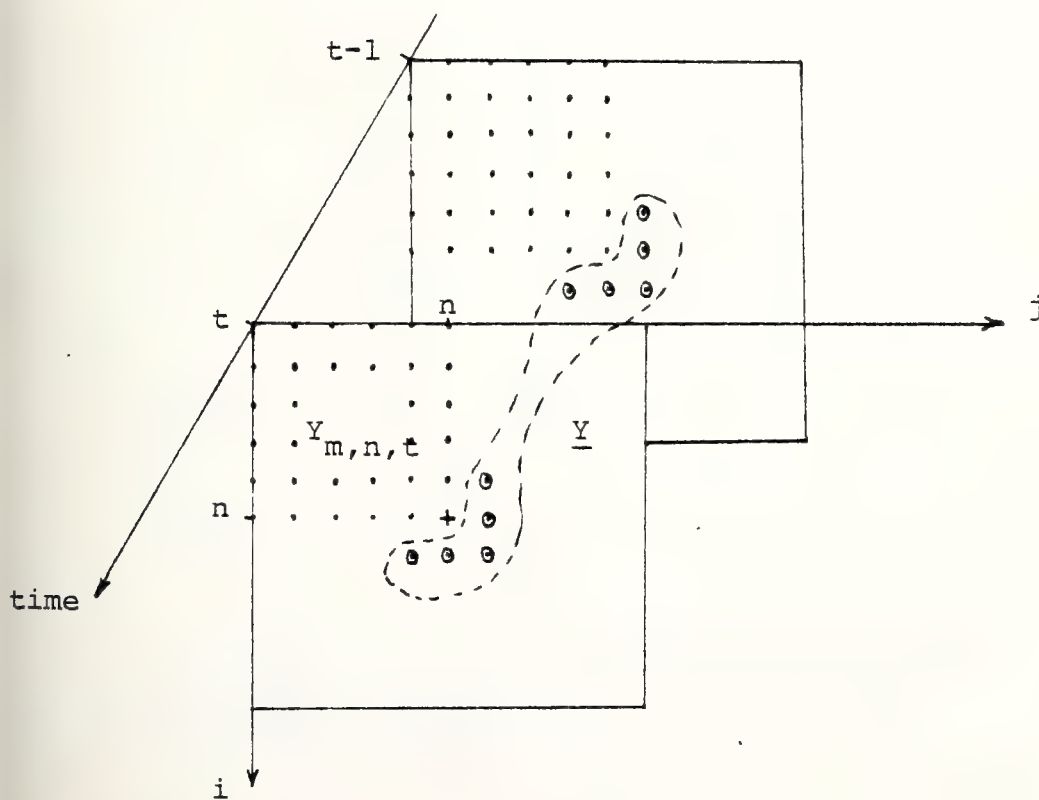


Figure 5.17

Three-Dimensional Hybrid Smoother

The design of this filter is similar to the hybrid smoother of the previous item, therefore, we present below the results.

The autocorrelation function used is

$$R(i,j) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \quad (5.10)$$

The predictor is

$$\hat{X}(m,n) = \underline{h}^T \underline{g}$$

where

$$\underline{g} = \begin{bmatrix} \hat{X}_p(m,n) \\ y(m-1,n+1) \\ y(m-1,n+2) \\ y(m-2,n+1) \\ y(m-2,n+2) \end{bmatrix} \quad \underline{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix}$$

The weights \underline{h} are given by

$$\underline{h} = R_{gg}^{-1} R_{Xg}$$

Using the results of Chapter III we can compute all the elements of R_{gg} and R_{Xg} . Let's call:

$$E[\hat{X}_p^2(m,n)] = \sigma_p^2 \quad (5.11)$$

$$E[\hat{X}_p] = b_p$$

It can be verified that the cross-correlation between \hat{X}_p and \underline{y} can be expressed as a function of b_p , as follows:

$$\begin{aligned} E[\hat{X}_p y(m-1, n+1)] &= \rho_v \rho_h b_p \\ E[\hat{X}_p y(m-1, n+2)] &= \rho_v \rho_h^2 b_p \\ E[\hat{X}_p y(m-2, n+1)] &= \rho_v^2 \rho_h b_p \\ E[\hat{X}_p y(m-2, n+2)] &= \rho_v^2 \rho_h^2 b_p \end{aligned} \quad (5.12)$$

Using equations (5.10), (5.11) and (5.12), we can calculate the autocorrelation matrix R_{gg} and the correlation vector R_{Xg} :

$$R_{gg} = \begin{bmatrix} \sigma_p^2 & \rho_v \rho_h b_p & \rho_v \rho_h^2 b_p & \rho_v^2 \rho_h b_p & \rho_v^2 \rho_h^2 b_p \\ \sigma^2 + r & \rho_h \sigma^2 & \rho_v \sigma^2 & \rho_v \rho_h \sigma^2 & \\ \sigma^2 + r & \rho_v \rho_h \sigma^2 & \rho_v \sigma^2 & & \\ \sigma^2 + r & \rho_h \sigma^2 & & & \\ \sigma^2 + r & & & & \end{bmatrix} \quad (5.13)$$

$$R_{Xg} = \begin{bmatrix} b_p \\ \rho_v \rho_h \sigma^2 \\ \rho_v^2 \rho_h^2 \sigma^2 \\ \rho_v^2 \rho_h \sigma^2 \\ \rho_v^2 \rho_h^2 \sigma^2 \end{bmatrix}$$

C. THREE-DIMENSIONAL HYBRID FILTERS

In this section we present two hybrid filters applied to pictures sequenced in time. The first filter is a smoother to be used in image enhancement and the other is a predictor. The design of these filters is quite similar to those of the previous section.

1. Three-dimensional Hybrid Smoother

The filter structure is shown in figure 5.17. The recursive filter is that developed in Chapter IV. The estimate $\hat{X}(m,n,t)$ of the recursive filter is optimally combined with the set of measurements \underline{Y} , in order to have a better estimate $\hat{X}_1(m,n,t)$. The design is quite the same as in the two-dimensional case, thus we present only the results.

The autocorrelation used for the time-frame is

$$R(i,j,k) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \rho_t^{|k|} \quad (5.14)$$

The estimate is

$$\hat{X}(m,n,t) = \underline{h}^T \underline{g}$$

where:

$$\underline{g} = \begin{bmatrix} \hat{X}(m,n,t) \\ y(m+1,n-1,t) \\ y(m+1,n,t) \\ y(m+1,n+1,t) \\ y(m,n+1,t) \\ y(m-1,n+1,t) \\ y(m+1,n-1,t-1) \\ y(m+1,n,t-1) \\ y(m+1,n+1,t-1) \\ y(m,n+1,t-1) \\ y(m-1,n+1,t-1) \end{bmatrix} \quad \underline{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \\ h_{10} \\ h_{11} \end{bmatrix}$$

The weights \underline{h} are given by

$$\underline{h} = R_{gg}^{-1} R_{Xg}$$

Using the same notation as in Chapter IV, we have

$$E[\hat{X}^2(m,n,t)] = \sigma_e^2 \quad (5.15)$$

$$E[\hat{X}] = b$$

As before, the correlation between \hat{X} and the measurements \underline{Y} can be expressed as a function of b as follows

$$E[\hat{X}(m,n,t)Y(m+i,n+j,t+k)] = \rho_v^{|i|} \rho_h^{|j|} \rho_t^{|k|} b \quad (5.16)$$

where

$$\begin{aligned} (i,j,k) = & (1,-1,0), (1,0,0), (1,1,0), (0,1,0), \\ & (-1,1,0), (1,-1,-1), (1,0,-1), (1,1,-1), \\ & (0,1,-1), (-1,1,-1) \end{aligned}$$

Using equations (5.14), (5.15) and (5.16), we can calculate the autocorrelation matrix R_{gg} and the correlation vector R_{xg} , then the weights \underline{h} .

2. Three-dimensional Hybrid Predictor

This filter combines optimally the first-step prediction \hat{X}_p of the two-dimensional recursive filter [3] and the set of measurements \underline{Y} shown in figure 5.18. Since its design is quite similar to the two-dimensional case, we present only the results.

The autocorrelation function is given in (5.14) and the hybrid prediction is

$$\hat{X}(m,n,t) = \underline{h}^T \underline{g}$$

where:

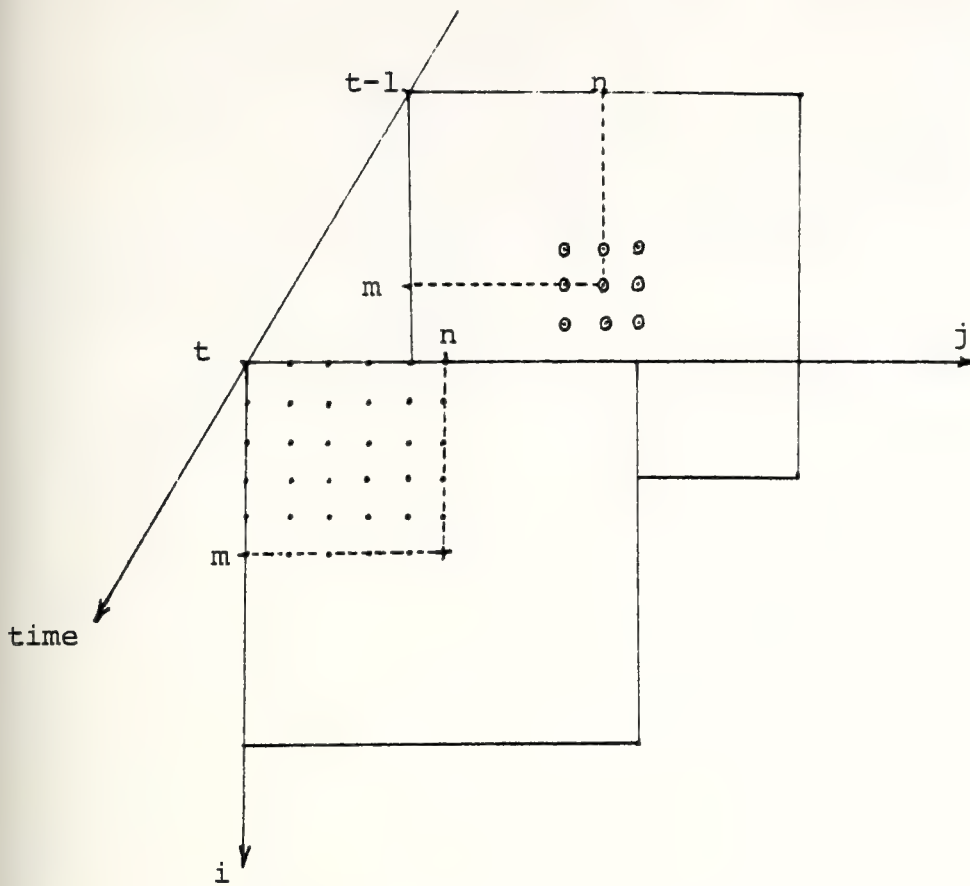


Figure 5.18
Three-dimensional hybrid predictor

$$\underline{g} = \begin{bmatrix} \hat{X}_p(m,n) \\ y(m-1,n-1,t-1) \\ y(m-1,n,t-1) \\ y(m-1,n+1,t-1) \\ y(m,n-1,t-1) \\ y(m,n,t-1) \\ y(m+1,n+1,t-1) \\ y(m+1,n-1,t-1) \\ y(m+1,n,t-1) \\ y(m+1,n+1,t-1) \end{bmatrix} \quad \underline{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \\ h_{10} \end{bmatrix}$$

The weights \underline{h} are given by

$$\underline{h} = R_{gg}^{-1} R_{Xg}$$

Similarly to the two-dimensional case:

$$E[\hat{X}_p^2] = \sigma_p^2 \quad (5.17)$$

$$E[X\hat{X}_p] = b_p$$

And the correlation between \hat{X}_p and \underline{y} is given by

$$E[\hat{X}_p(m,n,t)y(m+i,n+j,t-1)] = \rho_v^{|i|} \rho_h^{|j|} \rho_t b_p \quad (5.18)$$

where

$$(i,j) = (-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), \\ (1,-1), (1,0), (1,1)$$

The autocorrelation matrix R_{gg} and the correlation vector R_{Xg} can be calculated using equations (5.14), (5.17) and (5.18), then the weights \underline{h} are computed.

D. COMPARISON OF THE PREDICTORS

The predictors developed here will be used in the target detection problem addressed in Chapter VI. In this section we present some numerical results in order to evaluate its performance.

In Tables 5.3 through 5.5 the error variance of the recursive and hybrid predictors for several situations is shown. From these results we can see that the exploitation of the time correlation can be quite advantageous, mainly for the case of high correlation in time (table 5.3), where the error variance of the three-dimensional filters is around 50% of that given by the two-dimensional filters. We also can observe in the three tables that the 3D-hybrid filter is better than the 3D-recursive filter for observation noise variances above 0.2, although the former exploit only the time correlation with the previous frame. This result reinforces the basic idea of the hybrid filters, that is the use of all the neighbors of the pixel (m,n) , including those that are not causal.

Table 5.3

| Noise Variance | Error Variance | | | |
|--|-------------------|--------|-----------------|--------|
| | three-dimensional | | two-dimensional | |
| | recursive | hybrid | recursive | hybrid |
| 1.0 | .293 | .282 | .554 | .502 |
| 0.8 | .267 | .258 | .527 | .476 |
| 0.6 | .236 | .229 | .493 | .445 |
| 0.4 | .197 | .195 | .450 | .405 |
| 0.2 | .144 | .150 | .387 | .350 |
| Picture variance = 1.0 $\rho_v = \rho_h = 0.7$ $\rho_t = 0.95$ | | | | |

Table 5.4

| Noise Variance | Error Variance | | | |
|---|-------------------|--------|-----------------|--------|
| | three-dimensional | | two-dimensional | |
| | recursive | hybrid | recursive | hybrid |
| 1.0 | .351 | .303 | .427 | .364 |
| 0.8 | .324 | .283 | .400 | .340 |
| 0.6 | .293 | .260 | .367 | .312 |
| 0.4 | .251 | .231 | .324 | .277 |
| 0.2 | .191 | .192 | .263 | .227 |
| Picture variance = 1.0 $\rho_v = \rho_h = 0.8$ $\rho_t = 0.8$ | | | | |

Table 5.5

| Noise Variance | Error Variance | | | |
|---|-------------------|--------|-----------------|--------|
| | three-dimensional | | two-dimensional | |
| | recursive | hybrid | recursive | hybrid |
| 1.0 | .376 | .338 | .427 | .364 |
| 0.8 | .349 | .317 | .400 | .340 |
| 0.6 | .317 | .292 | .367 | .312 |
| 0.4 | .274 | .260 | .324 | .277 |
| 0.2 | .213 | .216 | .263 | .227 |
| Picture variance = 1.0 $\rho_v = \rho_h = 0.8$ $\rho_t = 0.7$ | | | | |

VI TARGET DETECTION

A. INTRODUCTION

In Chapter II we have developed models for images and for a sequence of images in time. Using such models we have constructed two and three dimensional recursive and hybrid filters to smooth out noise in images. In this chapter we are interested in the detection of targets using low contrast images. This problem is usually referred to as background suppression, where background means a kind of texture that predominates in the picture. In this context, the word target means the information that we are interested in extracting from the pictures. It may be a ship, an airplane or some kind of texture statistically different from that of the background.

Both textures, target and background, are modeled as in Chapter II, but it is assumed that less "a priori" knowledge is known about the target. The basic idea in detection is to use the prediction of the image filters previously developed. Knowing this prediction and the actual observation of the pixel we develop here, a decision rule to decide to which texture the pixel belongs.

B. TARGET AND BACKGROUND REGIONS

Figure 6.1 shows two distinct regions in the picture: background and target regions. In the visible spectrum what happens is a replacement process, we have background or target. On the other hand, in the infrared spectrum what seems to happen is some combination of the radiations from the target (e.g., a plane) and the background (e.g., clouds). To include these situations, let's assume that the gray level in the target region is given by

$$y(m,n) = \alpha x(m,n) + T(m,n) + v(m,n) \quad (6.1)$$

This model assumes a linear combination of target and background, normalized with respect to the target, and also consider an additive white observation noise $v(m,n)$, uncorrelated with both target and background textures. The background texture $x(m,n)$ is modeled, as in Chapter II, as a homogeneous random field with known mean and autocorrelation function and it is considered independent of the target texture. The target region is similarly modeled, but the amount of "a priori" knowledge, for most applications, may be restricted to the mean and/or variance.

The background region is modeled by

$$y(m,n) = x(m,n) + v(m,n) \quad (6.2)$$

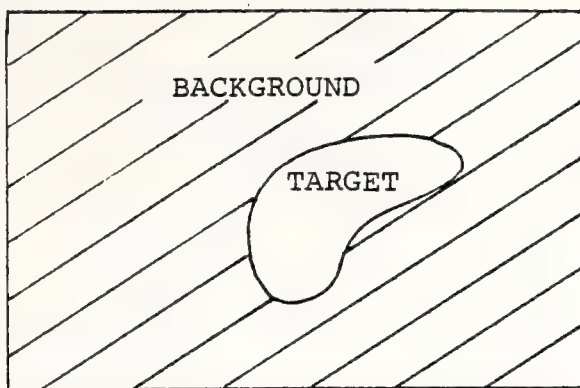


Figure 6.1
Background and Target Regions

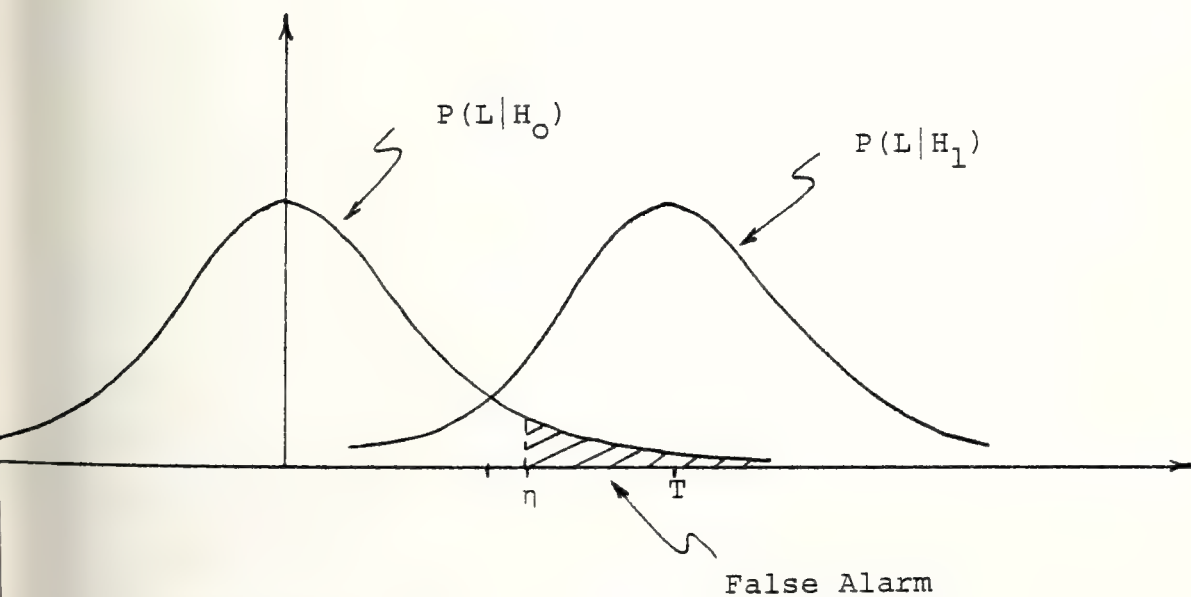


Figure 6.4
Conditional probability density function (case I)

The term $\alpha x(m,n)$ in (6.1) takes into account a possible correlation between the two regions. The weight α is a factor to be adjusted according to the specific nature of the picture. In the visible spectrum it seems reasonable to assume a replacement process, therefore $\alpha = 0$. In the infrared spectrum it may assume a constant value at the edges of the target and zero inside this region, provided we assume that the sensor has rejected the background.

C. LIKELIHOOD RATIO TEST

In this section we will construct a likelihood ratio to test the two hypotheses against a threshold. We will first consider the general case of deciding between two random fields, target and background, where the required "a priori" knowledge are the mean and autocorrelation function of the background and the mean and variance of the target and the measurement noise. After, we will work some special cases and derive its R.O.C. (Receiver Operating Characteristic). The detector developed here will be a Newman-Pearson detector [15], where the threshold is obtained from the R.O.C. subjected to the constraint that the probability of false alarm is less than some desired value.

The two hypotheses are:

Hypothesis H_0 - The measurement $y(m,n)$ is in the background region:

$$y|H_0 = y_0 = x + v$$

Hypothesis H_1 - The measurement $y(m,n)$ is in the target region:

$$y|H_1 = y_1 = \alpha x + T + v$$

The decision rule will be applied to each pixel using the observation $y(m,n)$ and the prediction of the pixel gray level $\hat{x}_p(m,n)$. Assume that the picture is being scanned row by row, from top to the bottom. For the sake of clarity, assume that the past pixels were correctly identified as belonging to the background region. The image filters developed before (recursive or hybrid) are used to give the prediction $\hat{x}_p(m,n)$. We are looking for a decision rule based only on $y(m,n)$ and $\hat{x}_p(m,n)$.

Let's define the likelihood ratio:

$$L(y, \hat{x}_p) = \ln \frac{p(\hat{x}_p y | H_1)}{p(\hat{x}_p y | H_0)} \quad (6.3)$$

where $p(\hat{x}_p y | \cdot)$ is the joint probability density function, conditioned to the hypothesis H_0 or H_1 .

Applying Bayes rule:

$$p(\hat{x}_p y | H_i) = p(\hat{x}_p | H_i) p(y | \hat{x}_p H_i) \quad i = 0, 1$$

Since the prediction \hat{x}_p is independent of the hypothesis:

$$p(\hat{x}_p | H_1) = p(\hat{x}_p | H_0)$$

Substituting these results into (6.3):

$$L(y, \hat{x}_p) = \ln \frac{p(y|\hat{x}_p H_1)}{p(y|\hat{x}_p H_0)} \quad (6.4)$$

At this point we will make the assumption that the conditional probabilities density functions in (6.4) are Gaussian.

We can write

$$y|H_0 = \hat{x}_p + v - e_p$$

$$y|H_1 = \alpha \hat{x}_p + v - \alpha e_p + T$$

Since \hat{x}_p is known, the conditional densities are functions of $(v - e_p)$ and $(v - \alpha e_p + T)$, where e_p is the prediction error. In figure (6.2) the histogram of $(v - e_p)$ is shown for the noisy image in figure (5.8), using the first-step predictor of the two-dimensional recursive filter [3]. This result shows a curve quite close to a normal distribution, and, therefore, validates the above assumption for $p(y|\hat{x}_p H_0)$ and also for $p(y|\hat{x}_p H_1)$, provided the target is deterministic.

Since the conditional probabilities density functions are assumed Gaussian, we need only to find its mean and variance. Before proceeding, let's first define some statistics for y and \hat{x}_p .

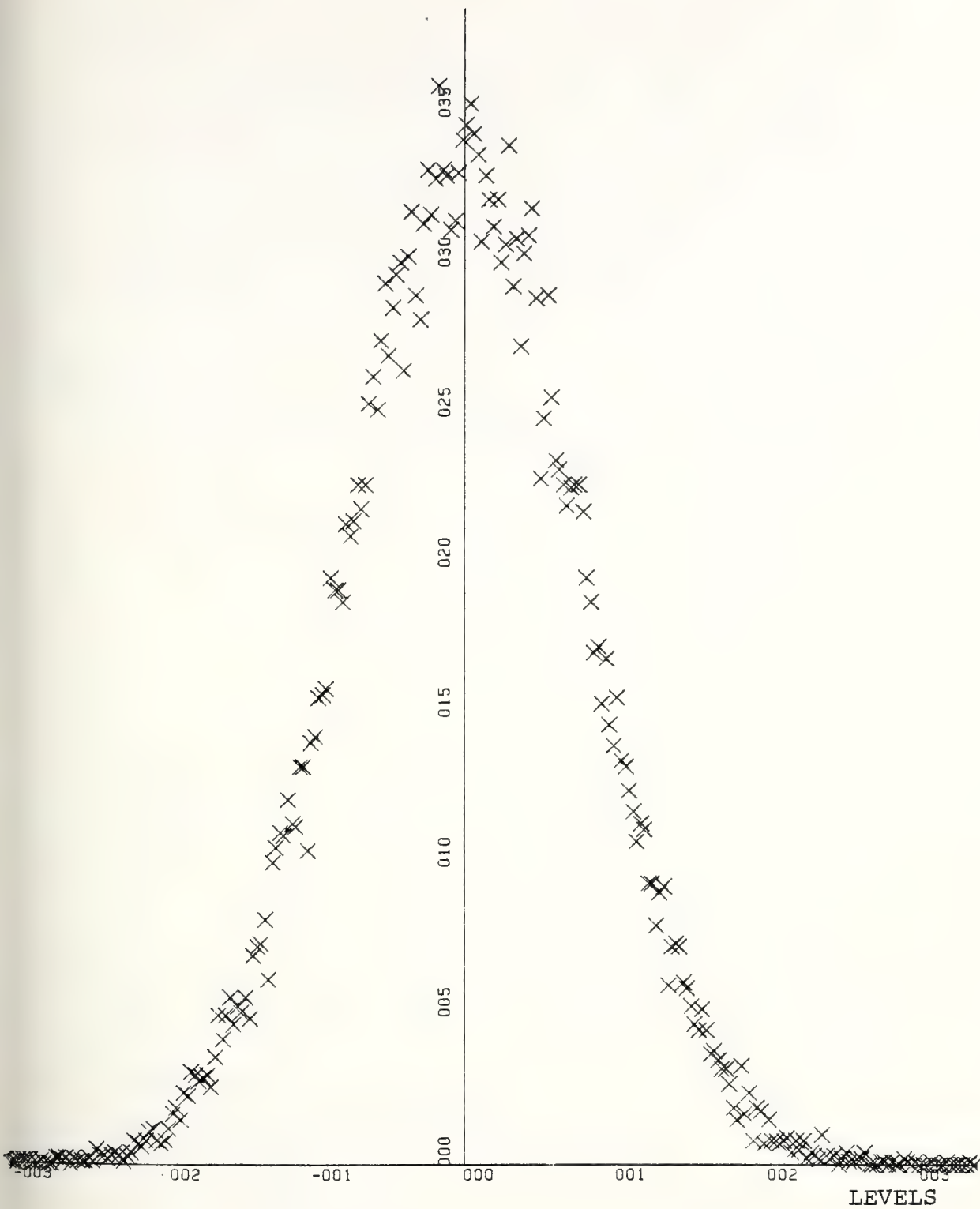


Figure 6.2

Histogram of $(v-e_p)$ for the image in figure (5.8)

(a) Means

To make the formulas more general, we consider here that the random fields $x(m,n)$ and $T(m,n)$ are not zero-mean. Since the fields are homogeneous, the mean is a constant and the same for all pixels. All the previous equations of the image filters remain valid, provided we change x to $(x - \bar{x})$ and y to $(y - \bar{x})$, where \bar{x} is the mean value of the field $x(m,n)$.

We will use the following means:

$$E(x) = \bar{x}$$

$$E(T) = \bar{T}$$

$$E(v) = 0$$

(6.5)

$$E(y|H_0) = \bar{y}_0 = \bar{x}$$

$$E(y|H_1) = \bar{y}_1 = \alpha\bar{x} + \bar{T}$$

$$E(\hat{x}_p) = E(x) = \bar{x}$$

(b) Variances

The variances are given by

$$V(x) = \sigma^2$$

$$V(x_p) = \sigma_p^2$$

$$V(T) = \sigma_T^2$$

$$V(v) = r$$

(6.6)

$$V(y|H_0) = \sigma_{Y_0}^2 = \sigma^2 + r$$

$$V(y|H_1) = \sigma_{Y_1}^2 = \alpha^2 \sigma^2 + \sigma_T^2 + r$$

(c) Covariances

We have the following covariances

$$\text{Cov}(v, x) = \text{Cov}(v, \hat{x}_p) = 0$$

$$\text{Cov}(v, y_i) = r, \quad i = 0, 1$$

$$\text{Cov}(\hat{x}_p, x) = b \quad (6.7)$$

$$\text{Cov}(\hat{x}_p, y_0) = \text{Cov}(\hat{x}_p, x) = b$$

$$\text{Cov}(\hat{x}_p, y_1) = \alpha \text{Cov}(\hat{x}_p, x) = \alpha b$$

(d) Coefficients of Correlation

The coefficients of correlation are

$$\rho_p = \frac{\text{Cov}(\hat{x}_p, x)}{\sigma_p \sigma} = \frac{b}{\sigma_p \sigma}$$

$$\rho_0 = \frac{\text{Cov}(\hat{x}_p, y_0)}{\sigma_p \sigma_{Y_0}} = \frac{b}{\sigma_p \sigma_{Y_0}} \quad (6.8)$$

$$\rho_1 = \frac{\text{Cov}(\hat{x}_p, y_1)}{\sigma_p \sigma_{y_1}} = \alpha \frac{b}{\sigma_p \sigma_{y_1}}$$

Now, returning to equation (6.4) and using the Gaussian pdf's, we have

$$L(y, \hat{x}_p) = \frac{[y - E(y|\hat{x}_p^{H_0})]^2}{V[y|\hat{x}_p^{H_0}]} - \frac{[y - E(y|\hat{x}_p^{H_1})]^2}{V[y|\hat{x}_p^{H_1}]} \quad (6.9)$$

where we have dropped some constants, since they are not important in the test against a threshold.

The regression of the mean of y on \hat{x}_p is

$$E[y|\hat{x}_p^{H_i}] = \bar{y}_i + \frac{\sigma_{y_i}}{\sigma_p} \rho_i (\hat{x}_p - \bar{\hat{x}}_p) \quad i = 0, 1 \quad (6.10)$$

Substituting equations (6.5) and (6.8) into (6.10) gives

$$E[y|\hat{x}_p^{H_0}] = (1-\gamma)\bar{x} + \gamma\hat{x}_p \quad (6.11)$$

$$E[y|\hat{x}_p^{H_1}] = \alpha(1-\gamma)\bar{x} + \bar{T} + \alpha\gamma\hat{x}_p$$

where we have defined:

$$\gamma = \frac{b}{\sigma_p^2} = \frac{\sigma}{\sigma_p} \rho_p \quad (6.12)$$

Having the means of the conditional pdf's, we find its variances:

$$\begin{aligned}
V[y|\hat{x}_p H_0] &= E\{[y_0 - E(y|\hat{x}_p H_0)]^2\} \\
&= E\{[x + v - (1-\gamma)\bar{x} - \gamma\hat{x}_p]^2\} \\
&= E\{[(x - \bar{x}) - \gamma(\hat{x}_p - \bar{x}) + v]^2\}
\end{aligned}$$

giving

$$V[y|\hat{x}_p H_0] = \sigma^2 + \gamma^2 \sigma_p^2 + r - 2\gamma \text{Cov}(\hat{x}_p, x)$$

Using equations (6.7) and (6.12) we have

$$\begin{aligned}
\text{Cov}(\hat{x}_p, x) &= \gamma \sigma_p^2 \\
V[y|\hat{x}_p H_0] &= \sigma^2 - \gamma \sigma_p^2 + r
\end{aligned} \tag{6.13}$$

$$\begin{aligned}
V[y|\hat{x}_p H_1] &= E\{[y_1 - E(y|\hat{x}_p H_1)]^2\} \\
&= E\{[\alpha x + T + v - \alpha(1-\gamma)\bar{x} - \bar{T} - \alpha\gamma\hat{x}_p]^2\} \\
&= E\{[\alpha(x - \bar{x}) + (T - \bar{T}) - \alpha\gamma(\hat{x}_p - \bar{x}) + v]^2\}
\end{aligned}$$

and

$$V[y|\hat{x}_p H_1] = \alpha^2 \sigma^2 - \alpha^2 \gamma^2 \sigma_p^2 + \sigma_T^2 + r \tag{6.14}$$

Substituting equations (6.11), (6.13) and (6.14) into (6.9) gives

$$L(y, \hat{x}_p) = \frac{[y - (1-\gamma)\bar{x} - \gamma\hat{x}_p]^2}{\sigma^2 - \gamma^2\sigma_p^2 + r} - \frac{[y - \alpha(1-\gamma)\bar{x} - \bar{T} - \alpha\gamma\hat{x}_p]^2}{\alpha^2(\sigma^2 - \gamma^2\sigma_p^2) + \sigma_T^2 + r}$$

Let's define:

$$M = (1-\gamma)\bar{x} \quad (6.15)$$

$$N = \sigma^2 - \gamma^2\sigma_p^2$$

$$\begin{aligned} L(y, \hat{x}_p) &= \frac{(y - \gamma\hat{x}_p - M)^2}{N+r} - \frac{(y - \alpha\gamma\hat{x}_p - \bar{T} - \alpha M)^2}{\alpha^2 N + r + \sigma_T^2} \quad (6.16) \\ &= \frac{(\alpha^2 N + r + \sigma_T^2)(y - \gamma\hat{x}_p - M)^2 - (N+r)(y - \alpha\gamma\hat{x}_p - \bar{T} - \alpha M)^2}{(N+r)(\alpha^2 N + r + \sigma_T^2)} \end{aligned}$$

The denominator is a positive constant, because

$$N = \sigma^2 - \gamma^2\sigma_p^2 = \sigma^2(1-\rho_p^2) \geq 0$$

Dropping the denominator and simplifying the numerator of (6.16), the likelihood ratio becomes

$$L(y, \hat{x}_p) = [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 - \delta[(y - \alpha\bar{x} - \bar{T}) - \alpha\gamma(\hat{x}_p - \bar{x})]^2 \quad (6.17)$$

where

$$\delta = \frac{N+r}{\alpha^2 N+r+\sigma_T^2} = \frac{(1-\rho_p^2)\sigma^2 + r}{\alpha^2 (1-\rho_p^2)\sigma^2 + r + \sigma_T^2} \quad (6.18)$$

Given the prediction \hat{x}_p and the measurement y , the decision rule consists in computing the likelihood ratio $L(\hat{x}_p, y)$ and compares it against a threshold η_1 if greater, we choose hypothesis H_1 , if not we choose hypothesis H_0 :

$$L(\hat{x}_p, y) \begin{cases} > \eta & \dots \text{Hypothesis } H_1 \\ < \eta & \dots \text{Hypothesis } H_0 \end{cases}$$

In figure 6.3 a block diagram of the estimator-detector filter is shown. It is assumed that the background predominates in the picture and the objective is the detection of a smaller texture that we call target. The background texture is modeled as in chapter II. To accomplish such modeling, however, a picture without target is needed. In an actual application it is realistic to assume that we have a typical realization of this process (background) before the target enters in the scene. For example, assume that we are dealing with a surveillance problem, where some area has been scanned many times looking for a very low contrast target. Therefore a recursive or hybrid filter can be designed to predict and estimate the

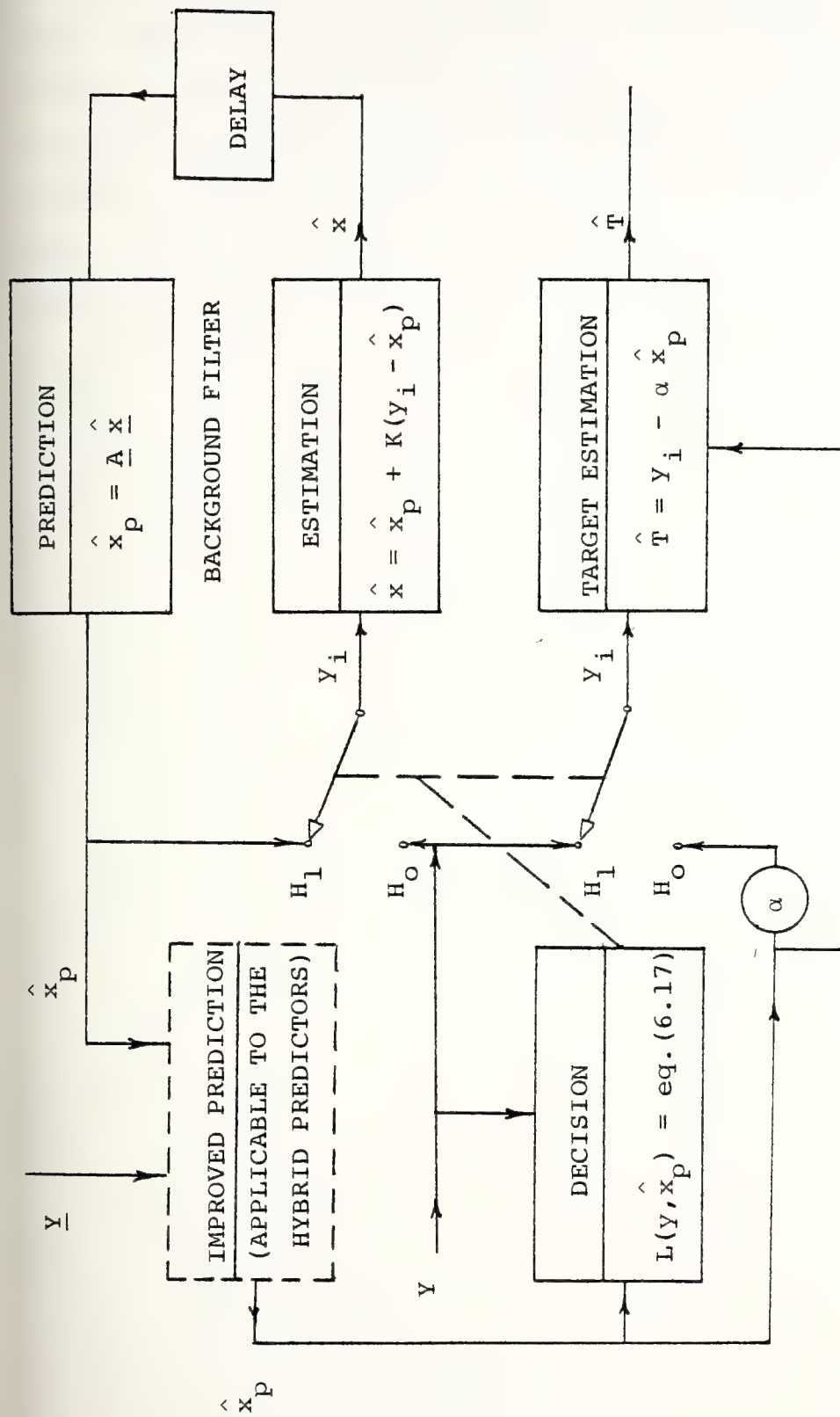


Figure 6.3
Estimator-detector filter

background. About the target we need only to have a good guess about the mean and variance of its gray level. In cases of very low contrast the mean may be assumed equal to that of the background. The variance, in many cases, may be assumed much smaller than that of the background. Also, in many cases, the observation noise is much smaller than the background.

The estimator-detector filter of figure 6.3 scans the image row by row from top to bottom. Assume that the previous pixels were correctly decided as belonging to the background. The decision box receives the measurement y and the prediction \hat{x}_p and computes the likelihood ratio $L(y, \hat{x}_p)$ from equation (6.17). If this ratio is less than a chosen threshold η , the pixel (m,n) is considered background and the observation y is used to find the background estimate $\hat{x}(m,n)$. If the ratio is greater than η , the pixel is considered as belonging to the target region. While in the target region the background filter cannot use the observation y , unless we have "a priori" knowledge about the target gray level. Since we are considering that the target region is small, the background filter may run in its prediction mode without much degradation, provided the dynamic model is accurate enough. Observe that both recursive and hybrid predictors can be used, the only difference being the box named "improved prediction" that applies only to the hybrid predictors. Since the three-dimensional predictors presented in chapter V, makes use of the measurements in

the previous frame, we can expect a good performance of those filters in the target region.

The gray level $T(m,n)$ of the target texture can be estimated using the background prediction \hat{x}_p and the observation $y|H_1$.

$$\hat{T}(m,n) = k_1 y_1 + k_2 \hat{x}_p$$

where

$$y_1 = y|H_1 = \alpha x + T + v$$

The estimation error is

$$e = T - \hat{T} = T - k_1(\alpha x + T + v) - k_2 \hat{x}_p \quad (6.19)$$

In order to choose an unbiased estimator:

$$E(e) = \bar{T} - k_1(\alpha \bar{x} + \bar{T}) - k_2 \bar{x} = 0$$

$$k_2 = \frac{1}{\bar{x}} [\bar{T} - k_1(\alpha \bar{x} + \bar{T})] \quad (6.20)$$

Using equations (6.19) and (6.20) the weights k_1 and k_2 can be found using the minimum-mean-squared error criteria:

$$k_1 = \frac{\alpha \bar{x} \bar{T} (\sigma_p^2 - b) + \bar{x}^2 \sigma_T^2 + \bar{T}^2 \sigma_p^2}{\alpha^2 \bar{x}^2 (\sigma_p^2 + \sigma_p^2 - 2b) + 2\alpha \bar{x} \bar{T} (\sigma_p^2 - b) + \bar{x}^2 \sigma_T^2 + \bar{T}^2 \sigma_p^2 + \bar{x}^2 r} \quad (6.21)$$

Observe that if we neglect the observation noise ($r \doteq 0$) and the prediction error ($\hat{x}_p \doteq x, b \doteq \sigma_p^2 \doteq \sigma^2$), equations (6.21) and (6.20) become

$$k_1 \doteq 1, \quad k_2 \doteq -\alpha$$

thus:

$$\hat{T} \doteq y_1 - \alpha \hat{x}_p \quad (6.22)$$

Equation (6.22) is intuitively appealing, since it is exactly what we expect in the absence of observation noise and with an ideal predictor. The block diagram of figure (6.3) considers this simplified case.

D. SPECIAL CASES

In this section we will study some particular cases of detection. The detector developed in the previous section will be applied to simple cases of practical interest and its performance (R.O.C.) will be derived.

1. Case I: $\alpha = 1, \sigma_T^2 = 0$

In this case the target gray level is deterministic and immersed in the background. The target gray level T is an unknown constant and we also don't know the target shape or location in the picture. The objective is to suppress the background in order to enhance the target. As before, the recursive or hybrid filters are used to

predict the background \bar{x} and the detector, using this prediction and the observation y , decide whether the pixel (m,n) is target or background. The output is a binary image where 1 is target and 0 is background.

The likelihood ratio, equation (6.17), becomes

$$L(y, \hat{x}_p) = [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 - \delta[(y - \bar{x} - T) - \gamma(\hat{x}_p - \bar{x})]^2$$

where

$$\delta = 1$$

Thus

$$L(y, \hat{x}_p) = 2T[(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})] - T^2$$

Dropping the constants, this ratio becomes:

$$L(y, \hat{x}_p) = (y - \bar{x}) - \gamma(\hat{x}_p - \bar{x}) \quad (6.23)$$

where

$$\gamma = \frac{\sigma}{\sigma_p} \rho_p$$

Equation (6.23) is intuitively appealing, because it is exactly the residue of the recursive filters when $\gamma = 1$.

This would happen if the filters were optimum. Such a decision rule with $\gamma = 1$ was used in [5] as a generalization of the one-dimensional Kalman filtering technique. However, we have shown here that such an extension is not correct, because the filters are not optimum, and, consequently, the residue (or innovation process [16]) does not correspond to the likelihood ratio. However, it is a good approximation to make $\gamma \doteq 1$ for the recursive filters, in cases of low observation noise and highly correlated background. It can be verified that $\gamma = 1$ for the hybrid filters.

Now, we will make the assumption that y and \hat{x}_p have a joint Gaussian distribution in order to find the performance of the detector. Previously we had assumed that the conditional pdf's were normally distributed and we validated such assumption with experimental results, but this does not mean that the joint distribution is also Gaussian. However, it is analytically convenient here, in order to have simple results.

If y and \hat{x}_p are jointly Gaussian, the likelihood ratio L is also Gaussian and its pdf is

$$P_L(l) = \frac{1}{\sqrt{2\pi} \sigma_L} \exp\left[-\frac{(l - \bar{l})^2}{2 \sigma_L^2}\right] \quad (6.24)$$

where:

$$\bar{l} = \bar{y}_i - \bar{x} \quad i = 0, 1$$

$$\begin{aligned}
\sigma_L^2 &= \sigma_{y_i}^2 + \gamma^2 \sigma_p^2 - 2\gamma b \\
&= \sigma_{y_i}^2 - \gamma^2 \sigma_p^2 \\
&= \sigma^2 + r - \gamma^2 \sigma_p^2 \\
&= (1 - \rho_p^2) \sigma^2 + r
\end{aligned}
\tag{6.25}$$

The variable L has the same variance for both hypothesis H_0 and H_1 , only the means are different as given by

$$E(L|H_0) = \bar{\ell}_0 = (\overline{x+v}) - \bar{x} = 0$$

$$E(L|H_1) = \bar{\ell}_1(\overline{x+T+v}) - \bar{x} = T$$

In figure 6.4 the conditional pdf's of the likelihood ratio $L(y, \hat{x}_p)$ is shown.

The probability of false alarm is

$$\begin{aligned}
P_F &= \text{prob}(L|H_0 > \eta) \\
&= \frac{1}{\sqrt{2\pi} \sigma_L} \int_{\eta}^{\infty} \exp(-\ell^2/2\sigma_L^2) d\ell \\
&= \frac{1}{\sqrt{2\pi}} \int_{\eta/\sigma_L}^{\infty} \exp(-u^2/2) du
\end{aligned}$$

$$P_F = \text{erfc}(\eta/\sigma_L) \quad (6.26)$$

The probability of detection is

$$\begin{aligned} P_F &= \text{prob}(L|H_1 > \eta) \\ &= \frac{1}{\sqrt{2\pi} \sigma_L} \int_{\eta}^{\infty} \exp[-(\ell-T)^2/2\sigma_L^2] d\ell \\ P_D &= \text{erfc}\left(\frac{\eta-T}{\sigma_L}\right) \end{aligned} \quad (6.27)$$

where

$$\text{erfc}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} \exp(-u^2/2) du$$

The choice of the threshold can be done, for example, by specifying the maximum admissible probability of false alarm; in this case we have a Neyman-Pearson detector [15]. Another criteria is to minimize the overall probability of error:

$$\begin{aligned} P_E &= P(H_0)P(E|H_0) + P(H_1)P(E|H_1) \\ &= P(H_0)P_F + P(H_1)(1-P_D) \end{aligned}$$

If we know $P(H_0)$ and $P(H_1)$, an optimum threshold can be found that minimizes P_E . If this is not the case, the best we can do is to assume that the hypotheses are equally likely, $P(H_0) = P(H_1) = 1/2$. With this choice, it can be shown that the optimum threshold is $\eta = T/2$ and $P_E = P_F = 1 - P_D$.

The implementation of the estimator-detector for this case is similar to the one shown in figure 6.3. In this case, since we assume a target with constant gray level, and, also assuming neglectable observation noise, we might use the measurement in the recursive estimation, while the filter is in the target region, provided that \hat{T} is subtracted from $y|H_1$. The best estimate of T must occur at the edges of the target region, because there, the background prediction \hat{x}_p is the best.

2. Case II: $\alpha = 1, \sigma_T^2 \neq 0$

In this case the target gray level is a random variable with known mean \bar{T} and variance σ_T^2 and, as before, we don't know its shape or location in the picture.

The likelihood ratio, equation (6.17), becomes

$$\begin{aligned} L(y, \hat{x}_p) &= [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 - \delta [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x}) - \bar{T}]^2 \\ &= (1 - \delta) [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 + 2\delta \bar{T} [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})] - \delta \bar{T}^2 \end{aligned} \quad (6.28)$$

where

$$\delta = \frac{(1-\rho_p^2)\sigma^2 + r}{(1-\rho_p^2)\sigma^2 + r + \sigma_T^2}$$

and

$$1-\delta = \frac{\sigma_T^2}{(1-\rho_p^2)\sigma^2 + r + \sigma_T^2} > 0$$

Rearranging equation (6.28) in order to have a quadratic form, and observing that $(1-\delta)$ is positive:

$$\begin{aligned} \frac{L(y, \hat{x}_p)}{(1-\delta)} &= [(y-\bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 + 2\beta[(y-\bar{x}) - \gamma(\hat{x}_p - \bar{x})] \\ &\quad + \beta^2 - \beta(\beta + \bar{T}) \\ &= [(y-\bar{x}) - \gamma(\hat{x}_p - \bar{x}) + \beta]^2 - \beta(\beta + \bar{T}) \end{aligned}$$

Dropping the constants, this ratio becomes

$$L(y, \hat{x}_p) = [(y-\bar{x}) - \gamma(\hat{x}_p - \bar{x}) + \beta]^2 \quad (6.29)$$

where

$$\beta = \bar{T}[(1-\rho_p^2) \frac{\sigma^2}{T} + \frac{r}{T}]$$

$$\gamma = \frac{\sigma}{\sigma_p} \rho_p$$

To calculate the performance of the detector, we make again the assumption that y and \hat{x}_p have a joint Gaussian distribution.

Define the random variable Z :

$$Z = (y - \bar{x}) - \gamma(\hat{x}_p - \bar{x}) + \beta$$

Since y and \hat{x}_p have a bivariate normal distribution, Z is also normally distributed and its pdf is

$$p_Z(z) = \frac{1}{\sqrt{2\pi} \sigma_Z} \exp\left[-\frac{(z - \bar{z})^2}{2\sigma_Z^2}\right] \quad (6.30)$$

$$\bar{z} = \bar{y} - \bar{x} + \beta$$

$$\begin{aligned} \sigma_Z^2 &= \sigma_Y^2 + \gamma^2 \sigma_P^2 - 2\rho_i \gamma \sigma_Y \sigma_P \quad i = 0, 1 \\ &= \sigma_Y^2 - \gamma^2 \sigma_P^2 = \sigma_Y^2 - \rho_P^2 \sigma^2 \end{aligned}$$

Since $L = Z^2$, its pdf is

$$P_L(\ell) = \frac{1}{2\sqrt{\ell}} [P_Z(\sqrt{\ell}) + P_Z(-\sqrt{\ell})]$$

Thus

$$P_L(\ell) = \frac{1}{2\sigma_Z \sqrt{2\pi\ell}} \left\{ \exp\left[-\frac{(\sqrt{\ell} - \bar{z})^2}{2\sigma_Z^2}\right] + \exp\left[-\frac{(\sqrt{\ell} + \bar{z})^2}{2\sigma_Z^2}\right] \right\} \quad (6.31)$$

The mean and variance of $L(y, \hat{x}_p)$ can be computed using the Moment Generating Function of Z .

$$M_Z(t) = E(e^{zt}) = \exp[t\bar{z} + \frac{1}{2}\sigma_Z^2 t^2]$$

$$E(L) = E(Z^2) = M_Z^{II}(0) = \bar{z}^2 + \sigma_Z^2$$

$$V(L) = E(L^2) - E^2(L) = E(Z^4) - E^2(Z^2)$$

$$= M_Z^{IV}(0) - [M_Z^{II}(0)]^2$$

Therefore

$$E(L) = \bar{z}^2 + \sigma_Z^2$$

(6.32)

$$V(L) = 2\sigma_Z^2(\sigma_Z^2 + 2\bar{z}^2)$$

The distance between the means of $L(y, \hat{x}_p)$, under the hypothesis H_0 and H_1 , is calculated using (6.30) and (6.32):

$$d = E(L|H_1) - E(L|H_0)$$

$$E(L|H_1) = (\bar{T}+\beta)^2 + \sigma^2 + \sigma_T^2 + r + \gamma^2 \sigma_P^2 - 2\rho_1 \sigma_{Y_1} \sigma_P$$

$$= (\bar{T}+\beta)^2 + \sigma^2 + \sigma_T^2 + r - \gamma^2 \sigma_P^2$$

$$= (\bar{T}+\beta)^2 + \sigma_T^2 + (1-\rho_P^2) \sigma^2 + r$$

$$\begin{aligned}
E(L|H_0) &= \beta^2 + \sigma^2 + r + \gamma^2 \sigma_p^2 - 2\rho_0 \gamma \sigma_{y_0} \sigma_p \\
&= \beta^2 + \sigma^2 + r - \gamma^2 \sigma_p^2 \\
&= \beta^2 + (1-\rho_p^2) \sigma^2 + r
\end{aligned}$$

thus:

$$d = \bar{T}^2 + 2\beta\bar{T} + \sigma_T^2 \quad (6.33)$$

From equation (6.33) we can see that the distance between the means is non-zero even for a target with zero mean.

The probability of detection is

$$P_D = \text{prob}(\ell | H_1 > \eta) = \int_{\eta}^{\infty} p_L(\ell | H_1) d\ell \quad ,$$

Using equation (6.31):

$$\begin{aligned}
P_D &= \frac{1}{2} \int_{\eta}^{\infty} \frac{1}{\sigma_{z_1} \sqrt{2\pi\ell}} \exp \left[-\frac{(\sqrt{\ell} - \bar{z}_1)^2}{2\sigma_{z_1}^2} \right] d\ell \\
&\quad + \frac{1}{2} \int_{\eta}^{\infty} \frac{1}{\sigma_{z_1} \sqrt{2\pi\ell}} \exp \left[-\frac{(\sqrt{\ell} + \bar{z}_1)^2}{2\sigma_{z_2}^2} \right] d\ell \quad (6.34)
\end{aligned}$$

Using the complementary error function (erfc), equation (6.34) becomes

$$P_D = \text{erfc}\left[\frac{1}{\sigma_{z_1}}(\eta^{1/2} - \bar{z}_1)\right] + \text{erfc}\left[\frac{1}{\sigma_{z_1}}(\eta^{1/2} + \bar{z}_1)\right] \quad (6.35)$$

Similarly, the probability of false alarm is given by

$$P_F = \text{Prob}(\ell | H_0 > \eta) = \int_{\eta}^{\infty} P_L(\ell | H_0) d\ell \quad (6.36)$$

$$P_F = \text{erfc}\left[\frac{1}{\sigma_{z_0}}(\eta^{1/2} - \bar{z}_0)\right] + \text{erfc}\left[\frac{1}{\sigma_{z_0}}(\eta^{1/2} + \bar{z}_0)\right]$$

The values of \bar{z} and σ_z^2 are given by

$$\bar{z}_1 = \bar{T} + \beta = \bar{T}\left[1 + \frac{r}{\sigma_T^2} + (1-\rho_p^2) \frac{\sigma^2}{\sigma_T^2}\right] \quad (6.37)$$

$$\sigma_{z_1}^2 = (1-\rho_p^2)\sigma^2 + \sigma_T^2 + r$$

$$\bar{z}_0 = \beta = \frac{r}{\sigma_T^2} + (1-\rho_p^2) \frac{\sigma^2}{\sigma_T^2}$$

$$\sigma_{z_0}^2 = (1-\rho_p^2)\sigma^2 + r$$

To analyze the detector in terms of some meaningful parameters, let's define the ratios:

$$\text{BNR} = \frac{\sigma^2}{\bar{x}} (\text{background-to-noise ratio})$$

$$\text{TBR} = \frac{\bar{T}^2 + \sigma_T^2}{\sigma^2} (\text{target-to-background ratio})$$

$$\text{RTT} = \frac{\bar{T}^2}{\sigma_T^2}$$

Normalizing the equations with respect to the background ($\sigma^2 = 1$), equations (6.35) and (6.36) become

$$P_D = \text{erfc}\left[\sqrt{\frac{\eta}{a + \frac{\text{TBR}}{1 + \text{RTT}}}} - \sqrt{b + \text{RTT}}\right] \quad (6.39)$$

$$+ \text{erfc}\left[\sqrt{\frac{\eta}{a + \frac{\text{TBR}}{1 + \text{RTT}}}} + \sqrt{b + \text{RTT}}\right]$$

$$P_F = \text{erf}\left[\sqrt{\frac{\eta}{a}} - \sqrt{b}\right] + \text{erfc}\left[\sqrt{\frac{\eta}{a}} + \sqrt{b}\right] \quad (6.40)$$

where

$$a = \frac{1}{\text{BNR}} + (1 - \rho_p^2) \quad (6.41)$$

$$b = a \frac{\text{RTT}}{\text{TBR}} (1 + \text{RTT})$$

Using (6.39) and (6.40) we have calculated the R.O.C. (Receiver Operating Characteristic) of the detector. In figure 6.5 the effect of the target-to-background ratio

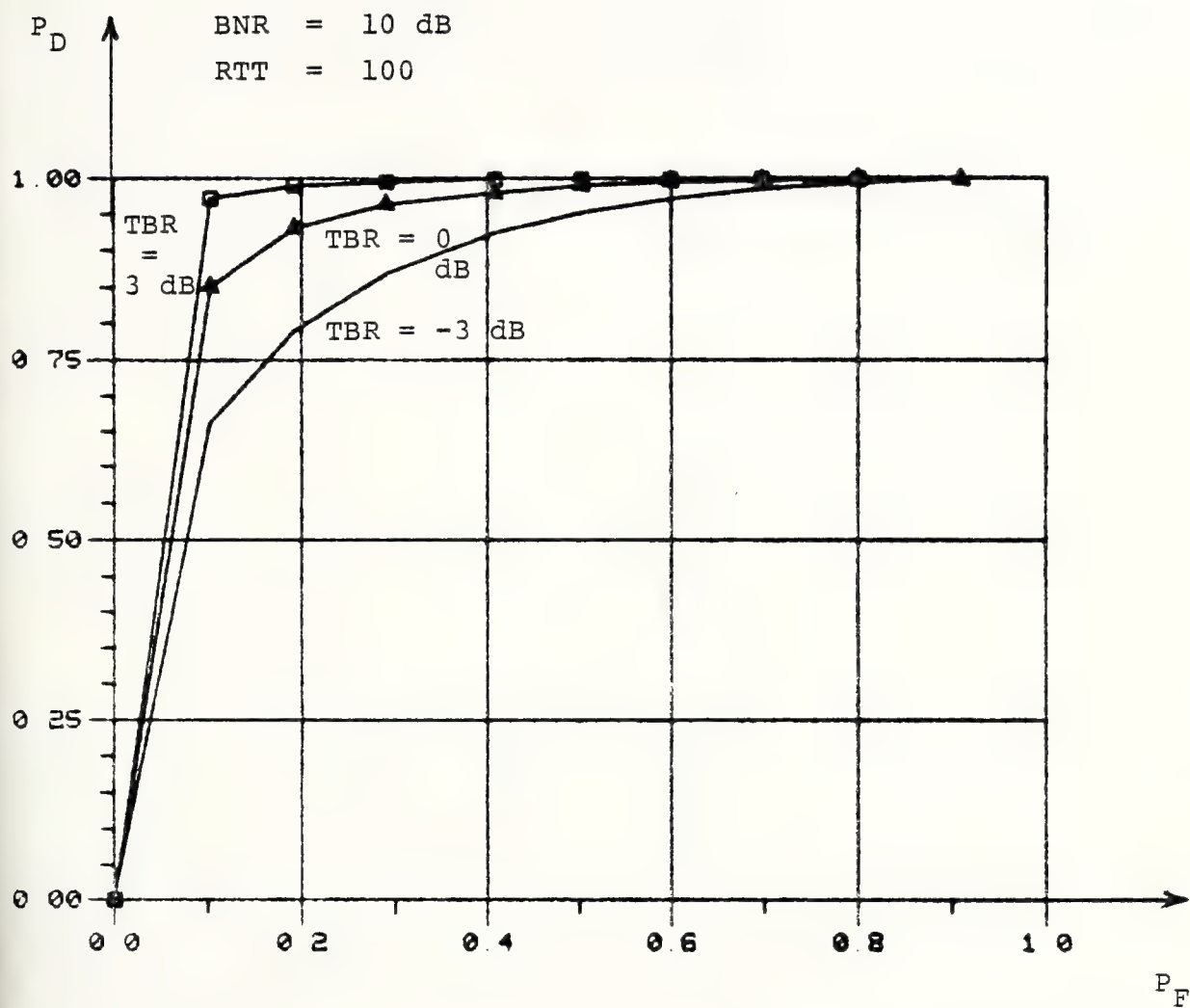


Figure 6.5

Curve of Performance changing the
target-to-background ratio (TBR)

(TBR) is shown. As expected, the detector performance is highly dependent of this ratio. For a probability of false alarm of 10% the probability of detection varies from 67% to 97% when TBR varies from -3 to 3 dB. In figure (6.6) the effect of the background-to-noise ratio is shown. Since this ratio is responsible for the accuracy of the background prediction (\hat{x}_p), it is also an important factor in the detection. From this figure we can see that, for a probability of false alarm of 10%, the probability of detection increased from 63% to 93% when the BNR varied from 10 to 30 dB.

In figure (6.7) the effect of the randomness of the target gray level, represented by the ratio RTT is shown. We can also observe a considerable improvement in the detection when this ratio is increased.

3. Case III: $\alpha = 0, \sigma_T^2 = 0$

In this case the target gray level is deterministic with known value, though its shape and location is unknown. The target and background regions are completely uncorrelated ($\alpha = 0$), thus it is the case, for example, of images in the visible spectrum where there is target or background, but not the addition of these textures.

The likelihood ratio, equation (6.17), for this case, becomes

$$L(y, \hat{x}_p) = [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 - \delta[y - T]^2 \quad (6.42)$$

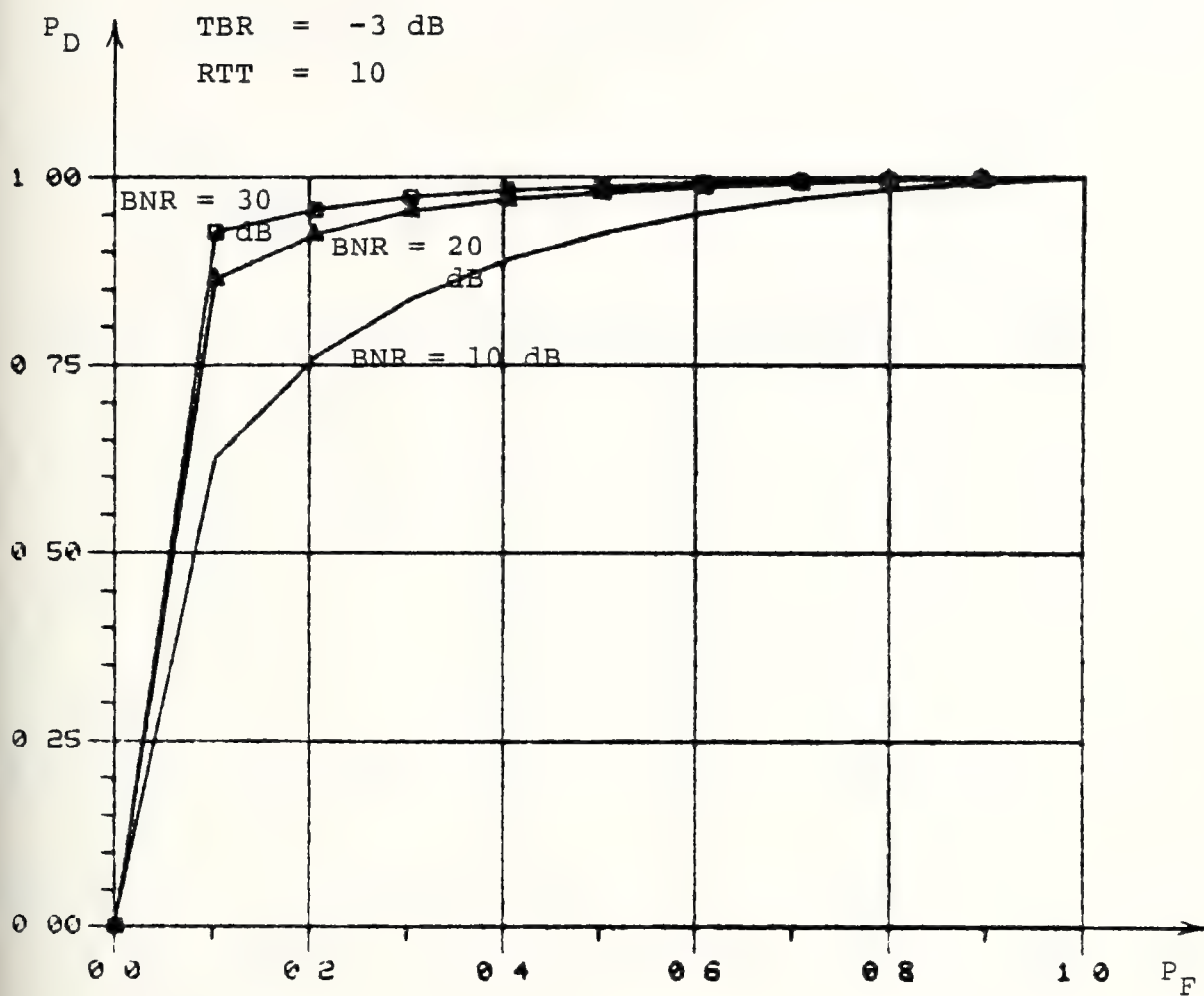


Figure 6.6

Curves of performance changing the background-to-noise ratio (BNR)

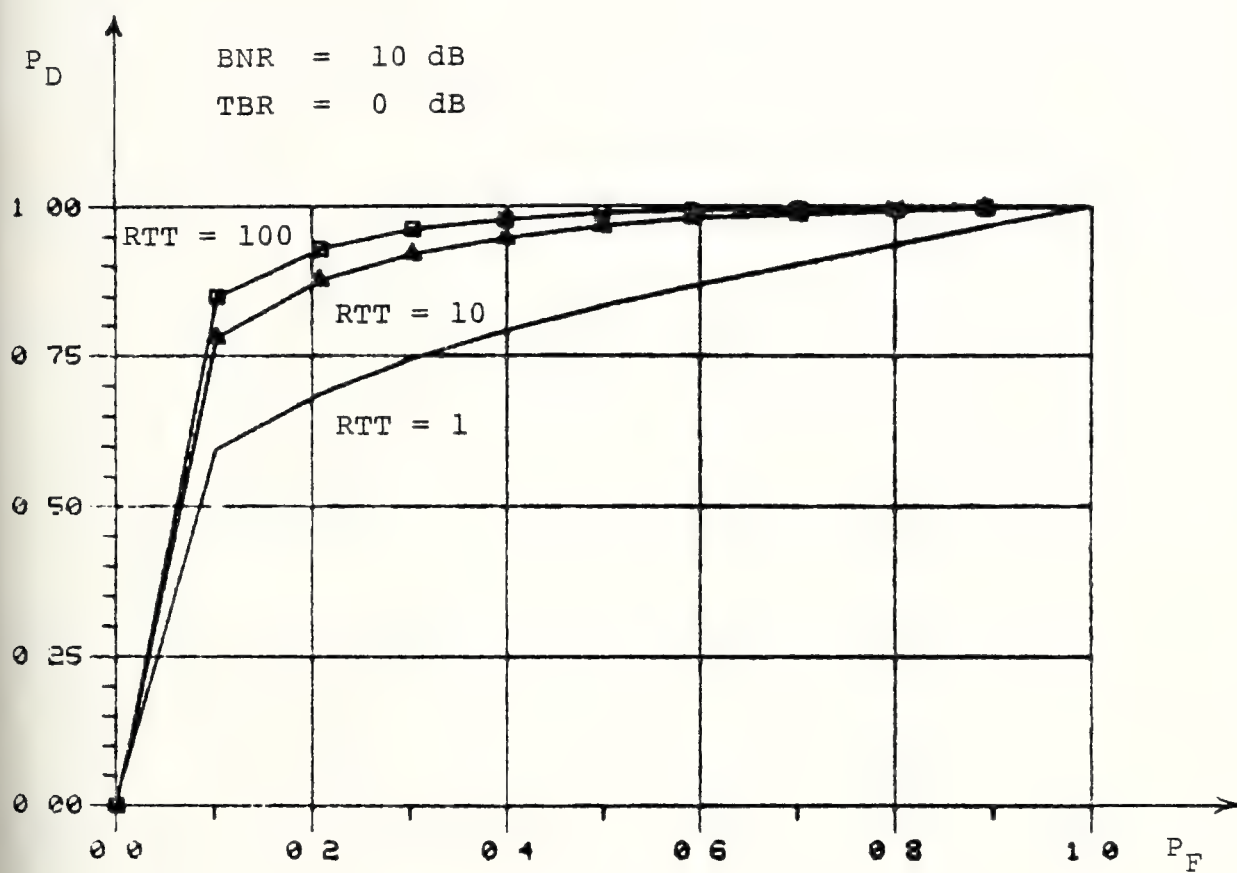


Figure 6.7

Curves of performance changing the ratio RTT

$$\delta = \frac{1}{r}(\sigma^2 - \gamma \sigma_p^2 + r) = \frac{1}{r}(1 - \rho_p^2) \sigma^2 + 1$$

Let's find the distance between the means, under the two hypotheses H_0 and H_1 :

$$L(Y, \hat{x}_p) = (Y - \bar{x})^2 + \gamma^2 (\hat{x}_p - \bar{x})^2 - 2\gamma (\hat{x}_p - \bar{x})(Y - \bar{x}) - \delta (Y - T)^2$$

$$Y|H_0 = x + v$$

$$E(L|H_0) = \sigma^2 + r + \gamma^2 \sigma_p^2 - 2\gamma b - \delta E[(Y - \bar{x}) - (T - \bar{x})]^2$$

$$= \sigma^2 + r - \gamma^2 \sigma_p^2 - \delta (\sigma^2 + r) - \delta (T - \bar{x})^2$$

$$E(L|H_0) = (1 - \delta)(\sigma^2 + r) - \gamma^2 \sigma_p^2 - \delta (T - \bar{x})^2 \quad (6.43)$$

$$Y|H_1 = T + v$$

$$E(L|H_1) = (T - \bar{x})^2 + (1 - \delta)r + \gamma^2 \sigma_p^2 \quad (6.44)$$

Subtracting (6.43) from (6.44) we find the distance

$$d = E(L|H_1) - E(L|H_0) \quad (6.45)$$

$$d = (1 + \delta)(T - \bar{x})^2 + (2\rho_p^2 + \delta - 1)\sigma^2$$

Observe that the distance is non-zero, even for targets with the same gray level as the mean of the background ($T = \bar{x}$).

The performance of the detector can be found, provided we assume, as before, a joint Normal distribution for y and \hat{x}_p , however it is not possible to have an elegant analytical solution as in the other cases.

4. Case IV: $\alpha = 0, \sigma_T^2 \neq 0$

This case might cover the model of some images with two quite distinct textures. A difficult problem, for example, is the restoration of pictures with two different regions, one is the sky and the other the sea. This is a typical case where we cannot model the image as a homogeneous random field. One approach to this problem is to model each texture, separately, as two homogeneous random fields, by the method of Chapter II, and design one image filter for each texture. To apply these filters, first we have to decide to which texture the pixel, to be processed, belongs. Such a decision can be made by the detector of the present case. In figure (6.8) the block diagram for this situation is shown. Following the same notation as before, we have a filter for the background x (one texture) and another filter for the target T (the other texture). The threshold decision selects the proper filter, while the other filter runs in its prediction mode.

The likelihood ratio for this case, equation (6.17), is

$$L(y, \hat{x}_p) = [(y - \bar{x}) - \gamma(\hat{x}_p - \bar{x})]^2 - \delta(y - \bar{T})^2 \quad (6.46)$$

$$\delta = \frac{(1-\rho_p^2)\sigma^2+r}{\sigma_T^2+r}$$

Let's compute the distance between the means:

$$E(L|H_0) = (1-\delta)(\sigma^2+r) - \gamma^2\sigma_p^2 - \delta(\bar{T}-\bar{x})^2$$

$$E(L|H_1) = (1-\delta)(\sigma_T^2+r) + \gamma^2\sigma_p^2 + (\bar{T}-\bar{x})^2$$

$$d = E(L|H_1) - E(L|H_0)$$

$$d = (1+\delta)(\bar{T} - \bar{x})^2 + (1-\delta)(\sigma_T^2-\sigma^2) + 2\rho_p^2\sigma^2 \quad (6.47)$$

Equation (6.47) shows clearly the differences between the two textures (mean and variance).

As in Case III, the performance of this detector can be found using the Gaussian hypothesis, however it is not possible to have an elegant analytical solution.

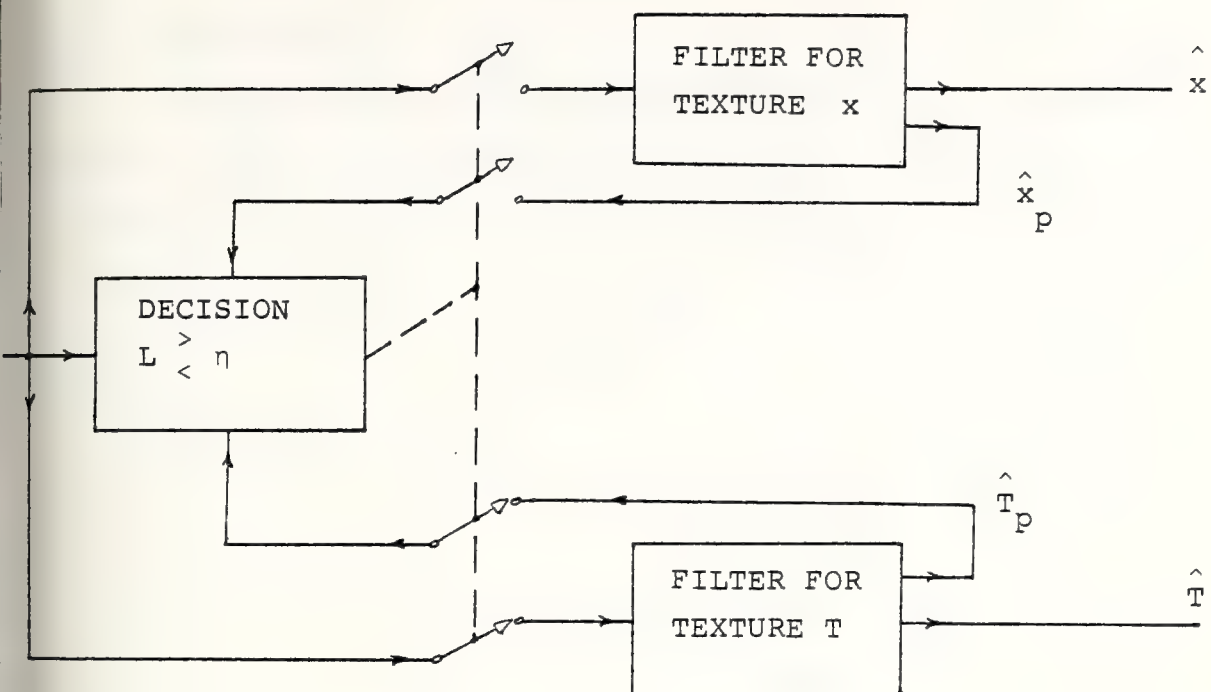


Figure 6.8

Estimator-detector filter for two textures

VII. TARGET TRACKING

A. INTRODUCTION

In the previous chapters we have developed statistical image filters that exploit spatial and/or temporal redundancies existing in the pictures. One application of those filters is the restoration of images contaminated with observation noise. In this application, the objective is to remove the observation noise, in order to recover the original image. Another application is to extract some desired information contained in the pictures, which we have called target. In this case, we want to suppress the rest of the information, which we have called background, in order to enhance the target.

The target detection and tracking problem, by means of pictures sequenced in time, may be divided into three phases:

- (a) Extraction of the targets from the background, creating a binary picture (1-target, 0-background).
- (b) Recognition of the target(s) that we are interested in tracking.
- (c) Track the target(s) from frame to frame.

This research addressed the first and third phases. The recognition problem, for our present purpose, will be considered as solved by a human operator, or, in the case of an automatic system, by some existing algorithms [18],

[19] of pattern recognition. The extraction problem was covered by the previous chapters.

In this chapter we address the tracking of targets from frame to frame. The basic idea is to model the target movement within the pictures, and to construct a Kalman filter to track the target centroid, using the image filters as the measurement device. Of course, others relevant points of the target might be tracked, in a similar way, in order to find, for example, the target attitude. The image filters referred to here are the three-dimensional recursive and hybrid predictors, presented in Chapters IV and V, respectively, and the detector developed in Chapter VI.

B. TARGET TRACKING FILTER

In this section we will construct a Kalman'filter to track the target centroid. Assume that we receive pictures at some specified rate, say 30 pictures/sec. Since the pictures have only two dimensions, the target-image dynamics is related only to the components of the actual movement that are parallel to the picture-plane (see figure 7.1). The effect of the movement perpendicular to the picture is to increase or decrease the size of the target-image. If the target has the same aspect at all angles (for example: a sphere), then it is possible to compute such component , using successive frames.

The target-image becomes a point-target at large distances, but, in general, it is a mass-target with some shape which changes when it maneuvers. Such change can have very interesting applications in the target tracking problem, since it gives useful information of target maneuvering.

In what follows, we model the movement of the target centroid in the plane of the picture and construct a Kalman filter to estimate its x and y coordinates, and also to predict its position in the next frame.

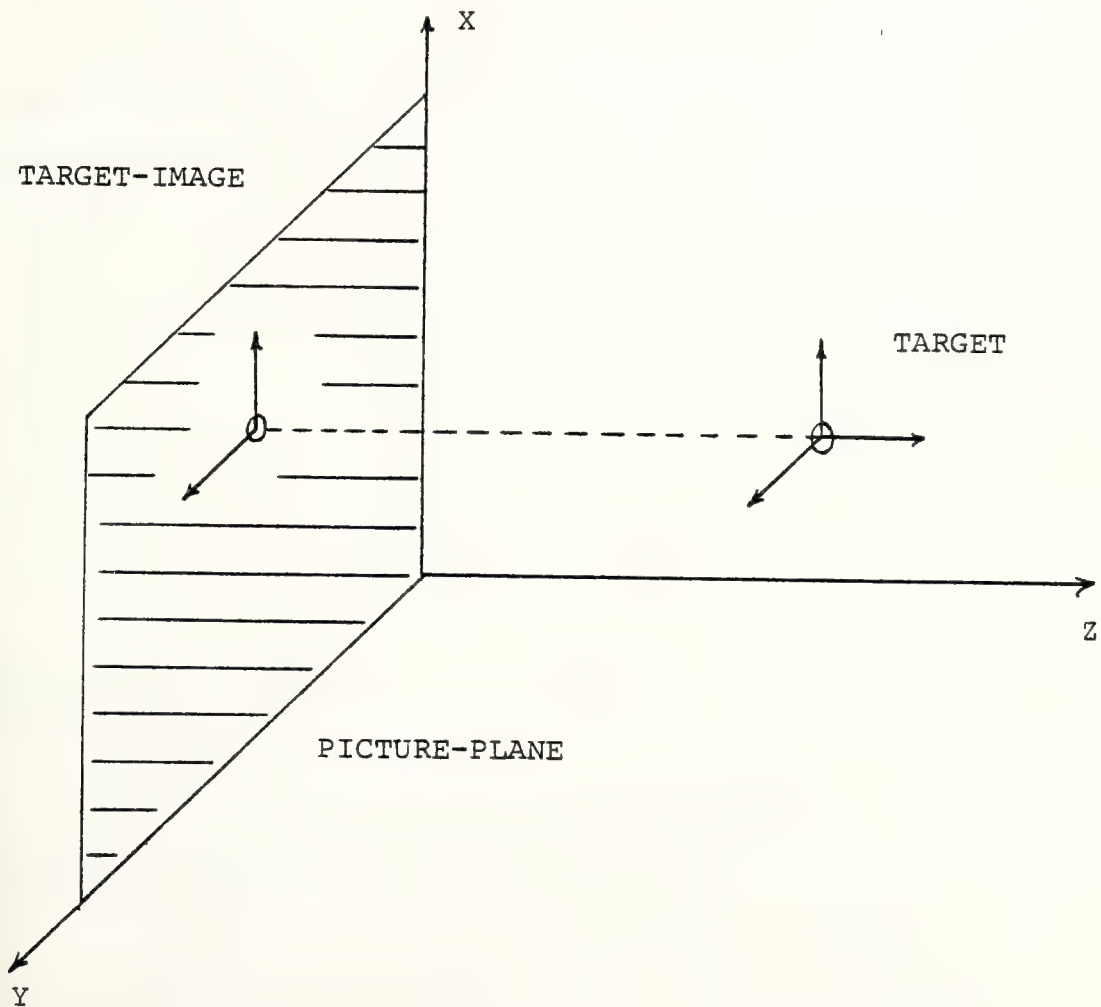


Figure 7.1
Target dynamics

1. Dynamic Model

Since we don't know the intention of the target driver, we will consider two random accelerations in the x and y directions, therefore the position of the target-image is given by

$$x(k+1) = x(k) + \dot{x}(k)T + \frac{1}{2}w_1(k)T^2 \quad (7.1)$$

$$y(k+1) = y(k) + \dot{y}(k)T + \frac{1}{2}w_2(k)T^2$$

Equation (7.1) assumes that the time separation T between frames is small enough such that we can consider that the acceleration and velocities are constants between frames.

The velocities are given by

$$\dot{x}(k+1) = \dot{x}(k) + w_1(k)T \quad (7.2)$$

$$\dot{y}(k+1) = \dot{y}(k) + w_2(k)T$$

where we have used in this derivation the simplified notation $x(k)$ for $x(kT)$.

To use state variable notation, define the state vector \underline{x} and the random forcing input \underline{w} :

$$\underline{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \quad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

In state space notation, equations (7.1) and (7.2) become

$$\begin{aligned} \underline{x}(k+1) &= \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \underline{x}(k) \\ &+ \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \underline{w}(k) \end{aligned} \quad (7.3)$$

The random forcing input \underline{w} is considered zero-mean white noise uncorrelated with \underline{x} , and has the covariance matrix below

$$E[\underline{w}(k)\underline{w}^T(k)] = \underline{Q} = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad (7.4)$$

2. The Observation Vector

The centroid of the target-image is measured by the image filters previously developed. The pixels are examined one by one and classified as target or background. After this classification the target centroid is computed by

$$\begin{aligned}x_m &= \frac{1}{S} \sum_{k=1}^N n_k T(m_k, n_k) , & S &= \sum_{k=1}^N T(m_k, n_k) \\y_m &= \frac{1}{S} \sum_{k=1}^N m_k T(m_k, n_k) & (7.5)\end{aligned}$$

where (m_k, n_k) are the coordinates of the pixels classified as target. If we create a binary picture with $T(m_k, n_k) = 1$, equations (7.5) become

$$\begin{aligned}x_m &= \frac{1}{N} \sum_{k=1}^N n_k \\y_m &= \frac{1}{N} \sum_{k=1}^N m_k\end{aligned} \tag{7.6}$$

Since the image filters are not perfect, some target-pixels are missed and some background-pixels are taken as targets. Therefore the measurement of the centroid has an error:

$$x_m = x + v_1 \quad (7.7)$$

$$y_m = y + v_2$$

where v_1 and v_2 are the errors in the x and y coordinates, respectively.

Let's analyze the errors v_1 and v_2 . First determine if they are uncorrelated. In figure 7.2 assume that the image filter missed the target-pixels in region A. The result is an error in the observation of the centroid. Both x and y coordinates will be in error. In other situations only one coordinate is wrong. In general, we can not draw any inference about one error by the knowledge of the other, therefore we will assume that v_1 and v_2 are uncorrelated.

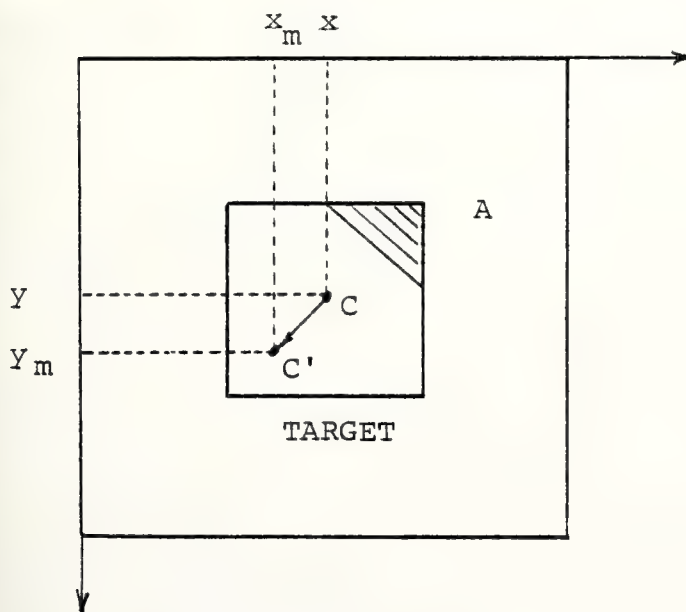


Figure 7.2
Measurement error

We must determine if the errors are correlated with the coordinates (x,y) . Since the background and the target gray levels are considered homogeneous random fields, there is no reason to think that the image filter will be more likely to make errors for some positions of the target within the frame, or at some special frames. On the other hand, the nature of the filter itself could be a source of correlation. Both image filters (recursive and hybrid) use recursion in space, therefore the background prediction is poor near the top left corner and also near the first row and column of all frames, because of the boundary transients of the filters. In the first frame the hybrid predictor does not make use of the time correlation in this frame. In the recursive predictor the first few frames have transients in recursion in time. The conclusion is that there are some positions (x,y) in the time-frame that have poor probability of detection. However, we have seen that the recursive filters converge very fast, therefore, we will neglect such correlation, since it occurs only in a very small region.

One must test to see if the errors are zero-mean. The detection error could happen at any pixel, but the bias is most likely in the direction that the picture is scanned (top left corner to right bottom corner), because when the filter is in the target region the background predictors, in general, can not use the observation $y(m,n)$, which

results in a degradation in the detection. However, we expect that such degradation can be neglected for practical applications where the target has small size and the background correlation is high. There are other schemes for processing that can be done to overcome this problem. One is for example, to process the pictures top-to-bottom and bottom-to-top, alternatively, from frame to frame.

After these considerations we conclude that it is reasonable to consider the errors v_1 and v_2 zero-mean white noise and uncorrelated with x and y . Such conclusions simplify considerably the following derivation and the result will be a very simple Kalman filter.

Define the measurement and error vectors as below

$$\underline{y} = \begin{bmatrix} x_m \\ y_m \end{bmatrix} \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (7.8)$$

In state space notation, equation (7.7) becomes

$$\underline{y}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underline{x}(k) + \underline{v}(k) \quad (7.9)$$

where $\underline{v}(k)$ is a zero-mean stationary random process, uncorrelated with $\underline{x}(k)$ and with the covariance matrix below

$$E[\underline{v}(k)\underline{v}^T(k)] = \underline{R} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \quad (7.10)$$

3. Kalman Filter

The dynamic model is

$$\underline{x}(k+1) = \Phi \underline{x}(k) + \Gamma \underline{w}(k) \quad (7.11)$$

where

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.12)$$

$$\Gamma = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \quad (7.13)$$

$$\underline{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \quad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (7.14)$$

The measurement model is given by

$$\underline{y}(k) = H \underline{x}(k) + \underline{v}(k) \quad (7.15)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7.16)$$

$$\underline{y} = \begin{bmatrix} x_m \\ y_m \end{bmatrix} \quad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (7.17)$$

The random forcing input $\underline{w}(k)$ and the measurement noise $\underline{v}(k)$ are considered white noise uncorrelated between themselves and also with the states. Its covariance matrices are given by

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \quad (7.18)$$

The Kalman filter equations are [17]:

$$\hat{\underline{x}}(k) = \underline{x}'(k) + G(k) [\underline{y}(k) - H\underline{x}'(k)] \quad (7.19)$$

where $\hat{\underline{x}}(k)$ is the estimated state vector and $\underline{x}'(k)$ is the predicted state vector given by

$$\underline{x}'(k) = \Phi \hat{\underline{x}}(k-1) \quad (7.20)$$

$G(k)$ is the filter gain, computed as below [17]:

$$G(k) = P'(k) H^T [H P'(k) H^T + R]^{-1} \quad (7.21)$$

$$P(k) = [I - G(k)H] P'(k) \quad (7.22)$$

$$P'(k+1) = \Phi P(k) \Phi^T + \Gamma Q \Gamma^T \quad (7.23)$$

where $P(k)$ is the covariance matrix of the state error and $P'(k)$ is the predicted covariance matrix of the state error.

The estimated state error vector is

$$\underline{e}(k) = \hat{\underline{x}}(k) - \underline{x}(k) \quad (7.24)$$

The covariance matrix of the state error is

$$P(k) = E[\underline{e}(k) \underline{e}^T(k)] \quad (7.25)$$

The predicted state error vector is

$$\underline{e}'(k) = \underline{x}'(k) - \underline{x}(k) \quad (7.26)$$

The predicted covariance matrix of the state error is

$$P'(k) = E[e'(k)e'^T(k)] \quad (7.27)$$

The gain $G(k)$ is computed by initialization of equation (7.21) with $P'(1)$ to have $G(1)$, then find $P(1)$ by equation (7.22) and $P'(2)$ by equation (7.23). Following this procedure the matrices $G(k)$, $P(k)$ and $P'(k)$ are calculated.

Since the x and y coordinates of the target centroid, as well as its measurement errors v_1 and v_2 , are uncorrelated, we can initialize equation (7.21) with the predicted covariance of the state error below

$$P'(1) = \begin{bmatrix} P'_{11}(1) & P'_{12}(1) & 0 & 0 \\ P'_{12}(1) & P'_{22}(1) & 0 & 0 \\ 0 & 0 & P'_{33}(1) & P'_{34}(1) \\ 0 & 0 & P'_{34}(1) & P'_{44}(1) \end{bmatrix} \quad (7.28)$$

With such initialization the matrices $G(k)$ and $P(k)$ have the forms:

$$G(k) = \begin{bmatrix} g_1(k) & 0 \\ g_2(k) & 0 \\ 0 & g_3(k) \\ 0 & g_4(k) \end{bmatrix} \quad (7.29)$$

$$P(k) = \begin{bmatrix} P_{11}(k) & P_{12}(k) & 0 & 0 \\ P_{12}(k) & P_{22}(k) & 0 & 0 \\ 0 & 0 & P_{33}(k) & P_{34}(k) \\ 0 & 0 & P_{34}(k) & P_{44}(k) \end{bmatrix} \quad (7.30)$$

Working out the equations (7.21) through (7.23)

the results are

$$g_1(k) = \frac{P'_{11}(k)}{R_1 + P'_{11}(k)}$$

$$g_2(k) = \frac{P'_{12}(k)}{R_1 + P'_{11}(k)}$$

(7.31)

$$g_3(k) = \frac{P'_{33}(k)}{R_2 + P'_{33}(k)}$$

$$g_4(k) = \frac{P'_{34}(k)}{R_2 + P'_{33}(k)}$$

$$P_{11}(k) = [1-g_1(k)]P'_{11}(k)$$

$$P_{12}(k) = [1-g_1(k)]P'_{12}(k)$$

$$P_{22}(k) = P'_{22}(k) - g_2(k)P'_{12}(k)$$

(7.32)

$$P_{33}(k) = [1-g_3(k)]P'_{33}(k)$$

$$P_{34}(k) = [1-g_3(k)]P'_{34}(k)$$

$$P_{44}(k) = P'_{44}(k) - g_4(k)P'_{34}(k)$$

$$P'_{11}(k+1) = P_{11}(k) + 2TP_{12}(k) + T^2P_{22}(k) + \frac{1}{4}T^4Q_1$$

$$P'_{12}(k+1) = P_{12}(k) + TP_{22}(k) + \frac{1}{2}T^3Q_1$$

$$P'_{22}(k+1) = P_{22}(k) + T^2Q_1$$

(7.33)

$$P'_{33}(k+1) = P_{33}(k) + 2TP_{34}(k) + T^2P_{44}(k) + \frac{1}{4}T^4Q_2$$

$$P'_{34}(k+1) = P_{34}(k) + TP_{44}(k) + \frac{1}{2}T^3Q_2$$

$$P'_{44}(k+1) = P_{44}(k) + T^2Q_2$$

Initializing the filter equation with the initial prediction $\underline{x}'(1)$, we have the estimated state vector

$$\begin{aligned}\hat{\underline{x}}(k) &= \underline{x}'(k) + g_1(k) [x_m(k) - x'(k)] \\ \hat{\dot{\underline{x}}}(k) &= \dot{\underline{x}}'(k) + g_2(k) [x_m(k) - x'(k)]\end{aligned}\tag{7.34}$$

$$\hat{y}(k) = y'(k) + g_3(k) [y_m(k) - y'(k)]$$

$$\hat{\dot{y}}(k) = \dot{y}'(k) + g_4(k) [y_m(k) - y'(k)]$$

The predicted state vector is

$$\begin{aligned}\underline{x}'(k+1) &= \hat{\underline{x}}(k) + T \hat{\dot{\underline{x}}}(k) \\ \dot{\underline{x}}'(k+1) &= \hat{\dot{\underline{x}}}(k) \\ y'(k+1) &= \hat{y}(k) + T \hat{\dot{y}}(k) \\ \dot{y}'(k+1) &= \hat{\dot{y}}(k)\end{aligned}\tag{7.35}$$

In figure 7.3 we show a flowchart for implementation of the filter.

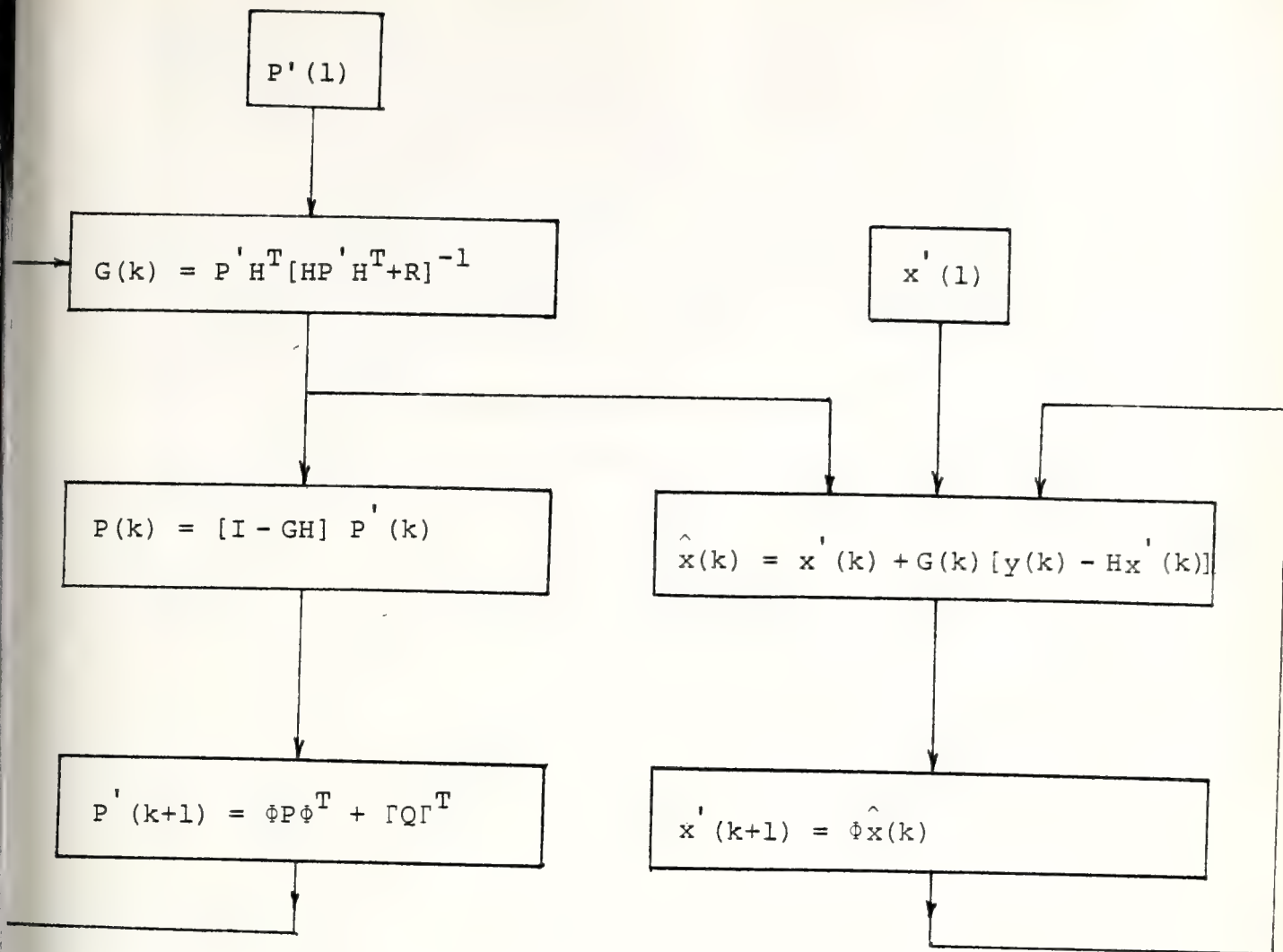


Figure 7.3
Kalman filter flowchart

C. FILTER APPLICATION

The tracking filter developed in this chapter is used, primarily, to provide the sensor servo with the predicted centroid coordinates for the next frame. Besides this application, the information of the predicted position of the target can be used:

- (a) To reduce the image processing,
- (b) To improve the performance of the target detector.

The reduction in the image processing can be achieved by delimiting to a small area to be processed, centered at the predicted position of the target centroid. Instead of processing, for example, a 100×100 picture we might work only in a 20×20 region, depending on the target size. With such reduction we could afford to make more processing in this smaller area, in order to improve the performance of the target detector.

Some examples of this extra processing are:

- (a) Scan the picture in two directions, one starting at the top left corner and the other at the bottom right corner. The results are two binary pictures (1-target, 0-background) with greater probability of error for the pixels located in the transient region of the recursion in space (see figure 7.4). One simple way to combine the two estimates is to pick the half of each picture opposite to the starting point, because, in this way we remove most of the unfavorable region of detection.

If more processing is allowable one might scan the picture four times starting at each corner and pick the quarter of each estimated picture opposite to the starting point. In this case the detection is completely free of the transient region.

- (b) In the detection of the target edges the direction of scanning the picture is very important. Since a good prediction of the background is needed and, in general, such prediction is degraded while in the target region, the result is that the detection of the edge at the left is better than the one at right (see figure 7.5). One way to overcome this problem is to use the estimator-detector only in favorable conditions. Assume that the prediction of the target centroid is good. Process first the left hand part of the picture until the line AB (see figure 7.5). Next, process the right hand part, starting at the top right corner. In this way, the filter will work in more favorable conditions, since it will face both left and right edges with good background predictions. The implementation of such a scheme may be simplified if we invert the columns of the right hand part before and after the processing. In this way, the filter estimates the left hand part and it is reinitialized at the boundary (line AB) to proceed to the estimation of the inverted right hand part. Therefore, the only additional process would be the required inversion.

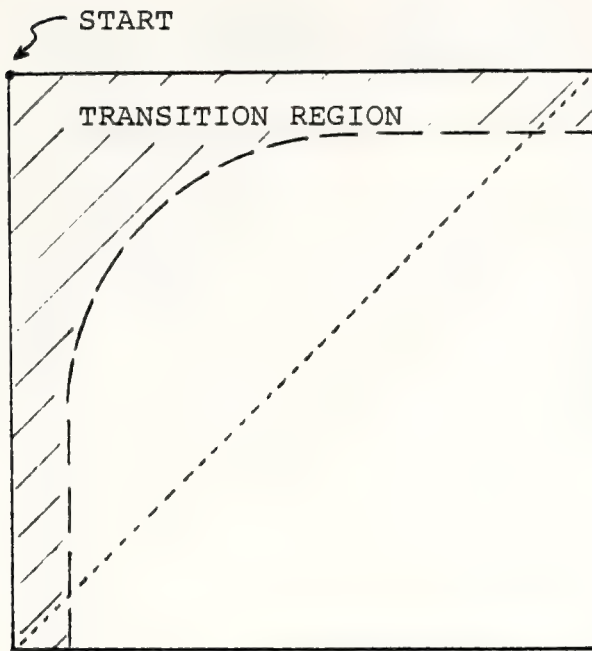


Figure 7.4
Transition Region

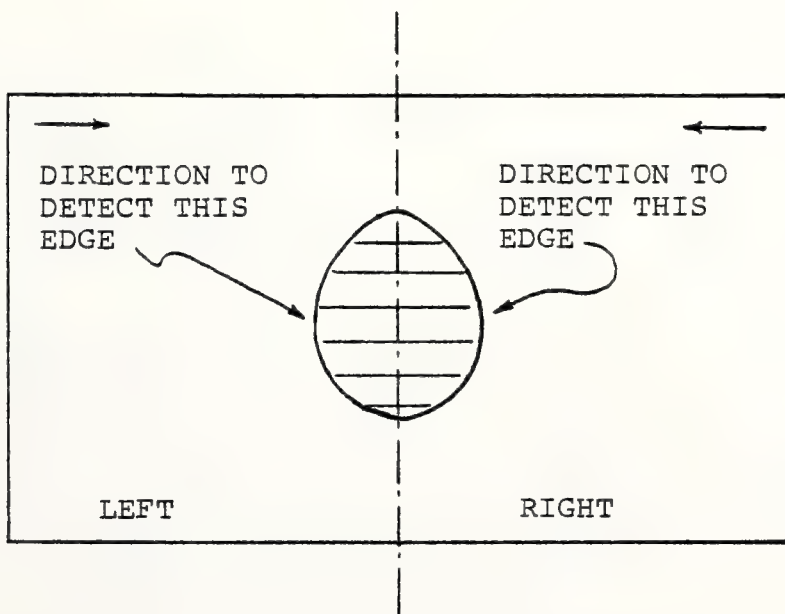


Figure 7.5
Edge Detection

VIII. COMPUTER SIMULATION

A. DESCRIPTION OF THE SYSTEM

A computer simulation of the target tracking and detection problem was implemented. The block diagram of the system is presented in figure 8.1. It is composed of three parts: time-frame generation, image processing and target tracking. The generator of the time-frame creates images at some specified rate and characteristics. The images contain the target and are contaminated with additive white Gaussian noise. The target has a random movement from frame to frame. The images are processed in two modes: search and window. The search mode is used to acquire the target. In this mode the whole picture is processed in order to extract the target from the background. It is assumed that a recognition phase exists to select the desired target. In this simulation we have used only one target to avoid such a phase. After the acquisition phase, the images are processed only in a window centered at the predicted position of the target centroid. The output of the image filter is the measurement of the coordinates (α_m, β_m) of the target centroid. The target tracking filter receives these measurements and updates the estimation of the target centroid $(\hat{\alpha}, \hat{\beta})$ and also

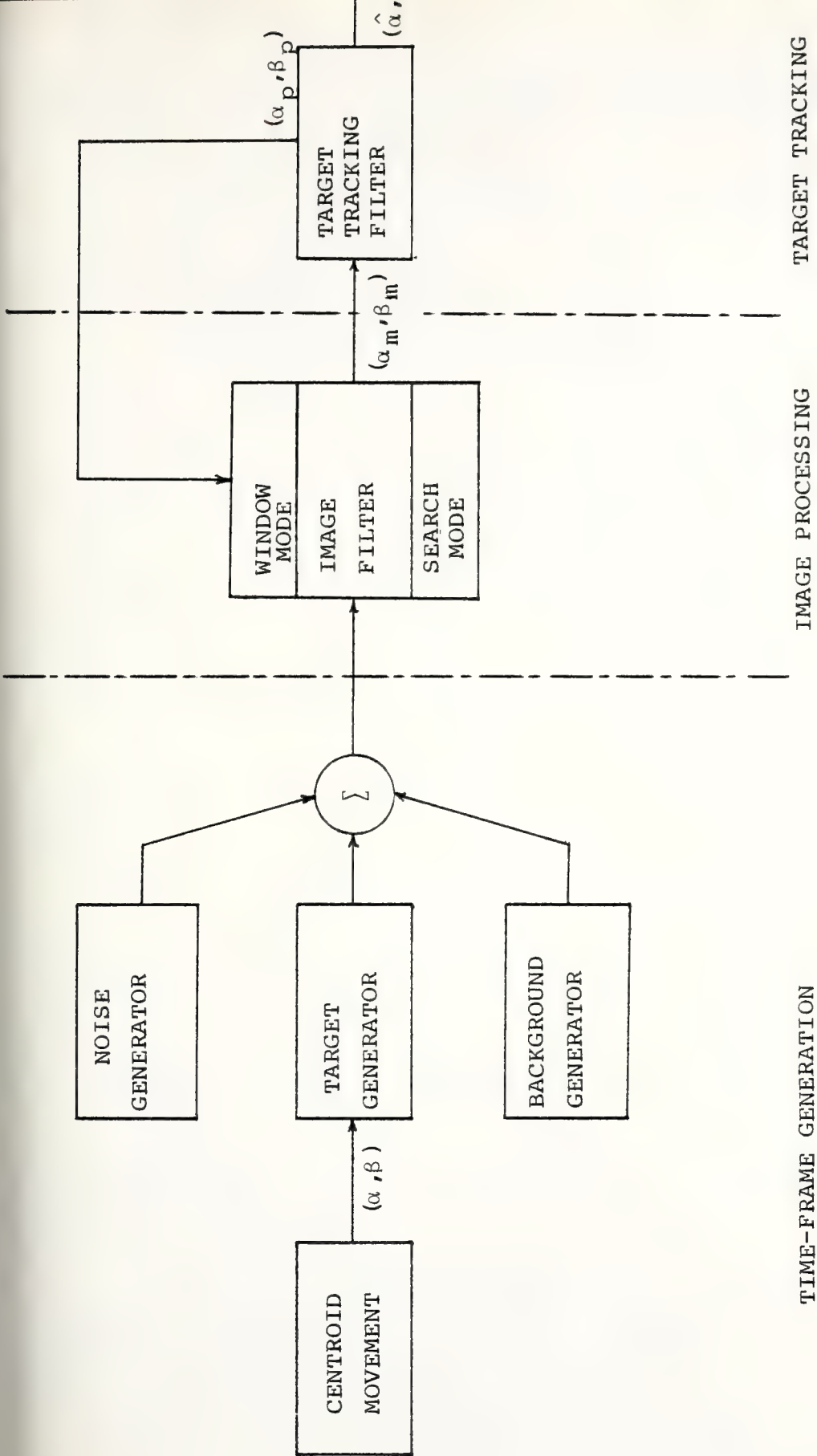


Figure 8.1
Block diagram of the simulation

computes the prediction for the next frame (α_p, β_p) which is feedback to the image filter.

More details about each block of figure 8.1 are given in the rest of this section.

1. Time-frame Generation

The background has the autocorrelation function:

$$R(i,j,k) = \sigma^2 \rho_v^{|i|} \rho_h^{|j|} \rho_t^{|k|} \quad i,j,k = 0, \pm 1, \pm 2, \dots$$

The dynamic model that generates a random field with such an autocorrelation function is

$$\underline{x}(m,n,t) = \underline{A} \underline{x} + w(m,n,t)$$

where:

$$\underline{A} = \begin{bmatrix} \rho_v & \rho_h \rho_t & -\rho_v \rho_h & -\rho_v \rho_t & -\rho_h \rho_t & \rho_v \rho_h \rho_t \end{bmatrix}$$

$$\underline{x}^T = \begin{bmatrix} x(m-1,n,t) & x(m,n-1,t) & x(m,n,t-1) & x(m-1,n-1,t) & x(m-1,n,t-1) & x(m,n-1,t-1) & x(m-1,n-1,t-1) \end{bmatrix}$$

$$x(m-1,n,t-1) \quad x(m,n-1,t-1) \quad x(m-1,n-1,t-1)]$$

$$E[w^2(m,n,t)] = Q = \sigma^2 (1 - \rho_v^2) (1 - \rho_h^2) (1 - \rho_t^2)$$

The block "Background generator" implements this dynamic model by driving the equation with zero-mean white Gaussian noise and variance Q.

The block "Target Generator" creates a binary image (T - target, 0 - background). The target has an arbitrary shape and it is centered at the location (x,y) given by the block "centroid movement". Such movement is generated using the dynamic model of equation (7.1).

The block "noise generator" creates an image with zero-mean white Gaussian noise at some specified variance.

Then, those three images are added in order to simulate the situation addressed in case I of chapter VI.

$$y|H_0 = x + v$$

$$y|H_1 = x + T + v$$

where:

x - background gray level

v - observation noise

T - target gray level

2. Image Processing

The three-dimensional recursive and hybrid filters were implemented in order to predict the background gray level \hat{x}_p . The decision rule using \hat{x}_p and the observation y is

$$L(y, \hat{x}_p) = y - \gamma \hat{x}_p$$

$$L(y, \hat{x}_p) \begin{cases} > \eta & \dots \text{Hypothesis } H_1 \\ < \eta & \dots \text{Hypothesis } H_0 \end{cases}$$

The block diagram of the combined estimator-detector filter is presented in figure 6.3, particularized to case I ($\alpha = 1, \sigma_T^2 = 0$).

The constant γ was defined (see equations 6.7 and 6.12) by

$$\gamma = \frac{\text{cov}(x, \hat{x}_p)}{\sigma_p^2} = \frac{b}{\sigma_p^2} = \frac{\sigma}{\sigma_p} \rho_p$$

where:

$$\begin{aligned} \sigma_p^2 &= E[\hat{x}_p^2] = E[\hat{x}_p(x + e_p)] \\ &= \text{cov}(x, \hat{x}_p) + E[\hat{x}_p e_p] \end{aligned}$$

For the recursive filter the prediction error e_p is not orthogonal to \hat{x}_p , because the filter is not optimum. However, the hybrid filter is such that the prediction error e_p is orthogonal to \hat{x}_p , therefore:

$$\gamma = \begin{cases} 1 & \dots \text{for the hybrid filter} \\ \frac{b}{\sigma_p^2} & \dots \text{for the recursive filter} \end{cases}$$

The threshold η is chosen such that the probability of a false alarm is below some desired value. From equation (6.26)

$$P_F = \operatorname{erfc} \left(\frac{\eta}{\sigma_L} \right)$$

$$\text{For } \eta = 5\sigma_L \quad P_F = 1.5 \times 10^{-6}$$

In the simulation the threshold can be specified by the operator or automatically set at the previous value

$$\eta = 5\sigma_L = 5 \sqrt{(1 - \rho_p^2) \sigma^2 + r} \quad (\text{see equation 6.25})$$

Observe that this criteria is completely independent of "a priori" knowledge of the target gray level.

We have defined the following ratios:

TBR_i = Target-to-background ratio at input

TBR_o = Target-to-background ratio at output

BNR = Background-to-noise ratio

P.G. = $\frac{TBR_o}{TBR_i}$ = Processing gain

$$TBR_i = \frac{T^2}{\sigma^2 + r} \quad (8.1)$$

$$TBR_o = \frac{E^2[L|H_1]}{\sigma_L^2} = \frac{T^2}{\sigma_L^2} \quad (8.2)$$

$$P.G. = \frac{\sigma^2 + r}{\sigma_L^2} = \frac{BNR + 1}{(1 - \rho_p^2) BNR + 1} \quad (8.3)$$

$$BNR = \frac{\sigma^2}{r}$$

From (8.3) we can see that one upper bound for P.G. is

$$P.G. \leq BNR + 1$$

Therefore, the observation noise has to be well below the background in order to have a substantial target enhancement. However, in many applications, the observation noise is really neglectable when compared with the background and, therefore, substantial target enhancement (or background suppression) can be achieved with such a method.

After the detection of each target-pixel, the centroid is computed by

$$\alpha_m = \frac{1}{N} \sum_{i=1}^N n_i$$

$$\beta_m = \frac{1}{N} \sum_{i=1}^N m_i$$

where (m_i, n_i) are the coordinates of the \underline{i} target-pixel.

3. Target Tracking Filter

The target tracking filter is the implementation of the Kalman filter constructed in Chapter VII. It receives the target centroid measurement (α_m, β_m) from the image filter and computes the estimated position $(\hat{\alpha}, \hat{\beta})$, as well as the predicted position in the next frame (α_p, β_p) . The filter gains are computed on-line according to the flowchart of figure 7.3.

B. SELECTED RESULTS

In this section we present some relevant results of the simulation. The pictures have the size 34 x 70, compatible with the Tektronics 4012 Display. The images are pictorially displayed with 8 levels of quantization obtained by superimposing some characters. The target has a "diamond" shape, but it can be easily changed to other shapes enclosed by a 7 x 7 matrix. In the tracking mode the image processing is only applied to a 20 x 20 window centered at the predicted centroid of the target.

1. Background Prediction

Since the decision rule is highly dependent on the background prediction, we have measured the prediction error for several situations and compared with the theoretical results. The results are presented in tables 8.1 through 8.3. Observe that the theoretical and experimental results have the same order of magnitude. We can also observe that the experimental results are always greater

TABLE 8.1

| NOISE VARIANCE | 3D-RECURSIVE PREDICTOR | | 3D-HYBRID PREDICTOR | |
|--|---------------------------|------------|------------------------|------------|
| | THEORY | EXPERIMENT | THEORY | EXPERIMENT |
| 1.0 | .293 | .306 | .282 | .360 |
| 0.8 | .267 | .280 | .258 | .332 |
| 0.6 | .236 | .255 | .229 | .308 |
| 0.4 | .197 | .216 | .195 | .266 |
| 0.2 | .144 | .166 | .150 | .206 |
| PICTURE VARIANCE = 1.0 $\rho_v = \rho_h = 0.7$ $\rho_t = 0.95$ | | | | |

TABLE 8.2

| NOISE VARIANCE | 3D-RECURSIVE PREDICTOR | | 3D-HYBRID PREDICTOR | |
|---|---------------------------|------------|------------------------|------------|
| | THEORY | EXPERIMENT | THEORY | EXPERIMENT |
| 1.0 | .351 | .364 | .303 | .353 |
| 0.8 | .324 | .343 | .283 | .329 |
| 0.6 | .293 | .316 | .260 | .295 |
| 0.4 | .251 | .266 | .231 | .269 |
| 0.2 | .191 | .207 | .192 | .217 |
| PICTURE VARIANCE = 1.0 $\rho_v = \rho_h = 0.8$ $\rho_t = 0.8$ | | | | |

TABLE 8.3

| NOISE VARIANCE | 3D-RECURSIVE PREDICTOR | | 3D-HYBRID PREDICTOR | |
|---|---------------------------|------------|------------------------|------------|
| | THEORY | EXPERIMENT | THEORY | EXPERIMENT |
| 1.0 | .376 | .388 | .338 | .378 |
| 0.8 | .349 | .353 | .317 | .352 |
| 0.6 | .317 | .337 | .292 | .330 |
| 0.4 | .274 | .294 | .260 | .295 |
| 0.2 | .213 | .231 | .216 | .244 |
| PICTURE VARIANCE = 1.0 $\rho_v = \rho_h = 0.8$ $\rho_t = 0.7$ | | | | |

than the theoretical ones. Such discrepancy is due to the fact that the theoretical values are computed for the steady-state condition, but the experimental values are the result of averaging over a small ensemble (20 frames) and small images (34 x 70), which means a reasonable influence of the transient of the image filters.

Therefore, we consider that these results are a reasonable validation for the theoretical methods of filter design and performance evaluation presented before.

2. Transition Region

To observe the transient of the image filters, we have applied the decision rule to a time sequence of images

(time-frame) without target. The threshold was set at $\eta = 0.35$. The characteristics of the images are:

$$\sigma^2 = 1, \quad \rho_v = \rho_h = 0.93, \quad \rho_t = 0.95, \quad \text{BNR} = 30 \text{ dB}$$

The results for the recursive filter are shown in figures 8.2 through 8.5. As mentioned before, the 3D-recursive filter gives degraded predictions at the first few frames and near the first row and column for all frames, due to the time and space recursion, respectively. Observe that in frame 2 (figure 8.2) the false alarms occurred mostly near the first column and row, but they can also be seen in the rest of the picture. In frame 10 (figure 8.4) the situation is better, because at the steady-state was reached in time, but the false alarms persist due to the transient in space. In chapter VII we have suggested some methods to reduce such undesirable effects of the transition region. In this simulation we have used a simple method that is to increase the threshold for the pixels at the first 5 rows and columns. The results of such a method are shown in figures 8.3 and 8.5. Observe the substantial improvement in both frames 2 and 10. The false alarms of frame 2 (figure 8.3) are basically due to the transient in time. The percentage of false alarm after frame 5, as in frame 10 (figure 8.5), is quite close to the theoretical value. Of course, the probability of detection is reduced in the transient region, due to the higher threshold, however this region is very small for the usual size of pictures (100 x 100).

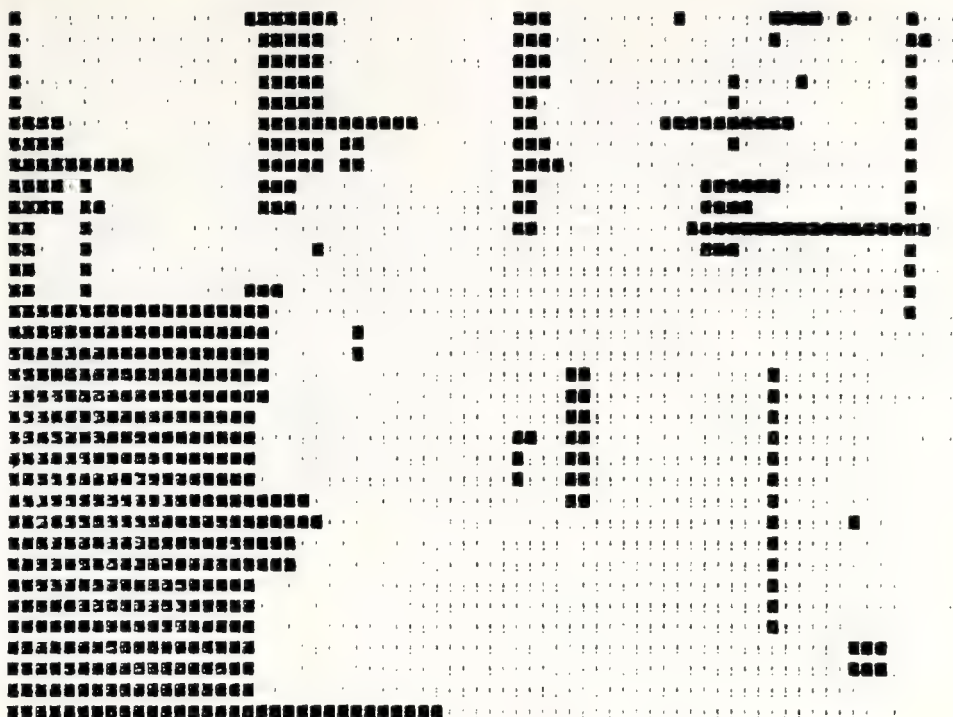


Figure 8.2 False alarms of the recursive filter at frame 2, using one threshold.

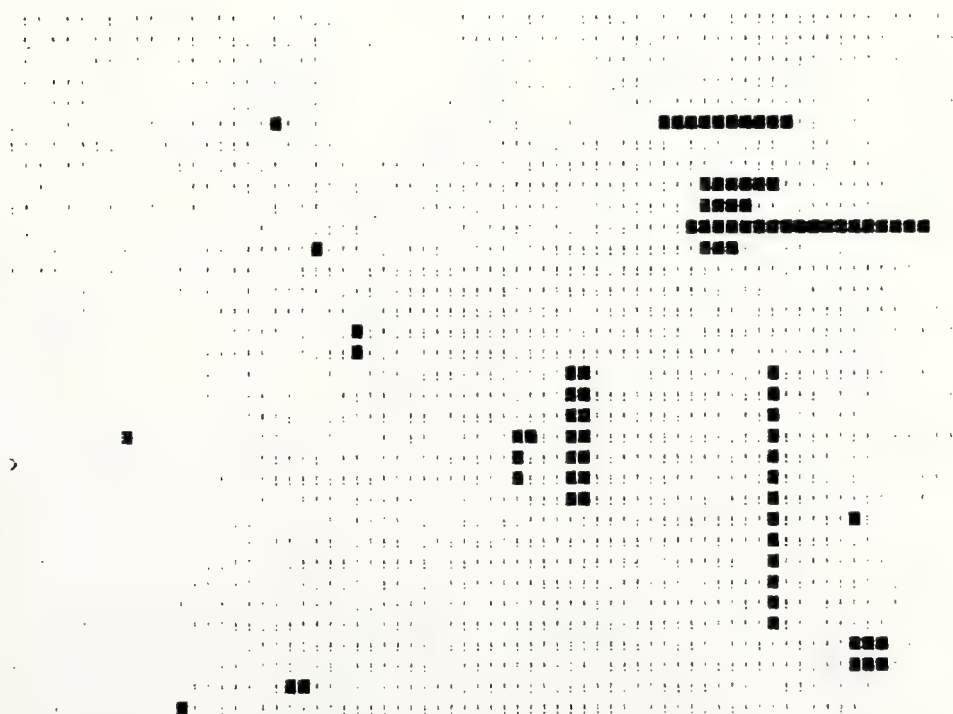


Figure 8.3 False alarms of the recursive filter at frame 2, using two thresholds.

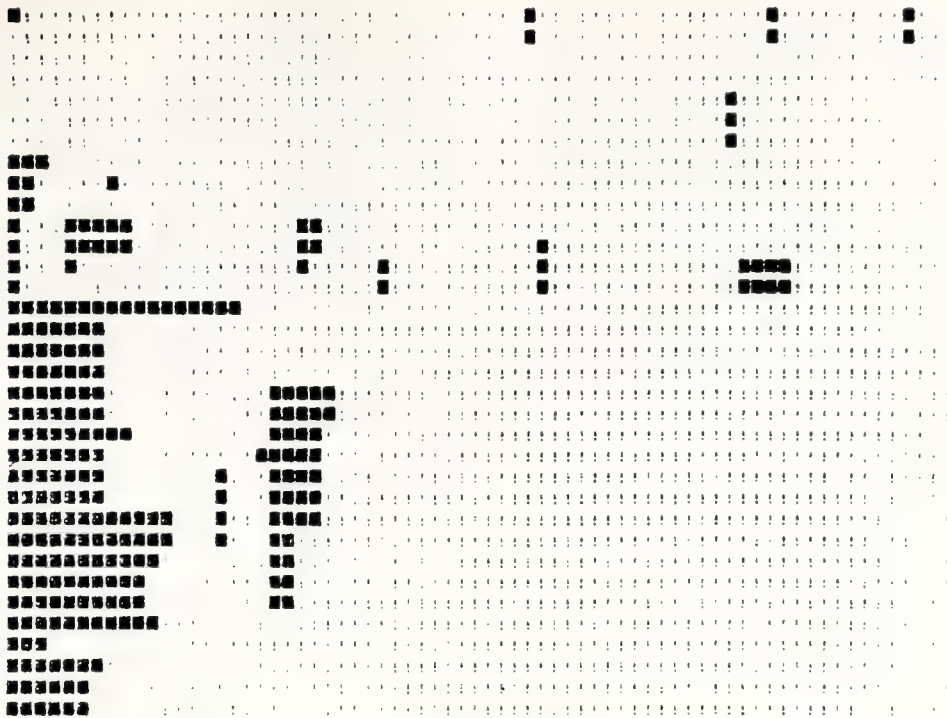


Figure 8.4 False alarms of the recursive filter at frame 10, using one threshold.

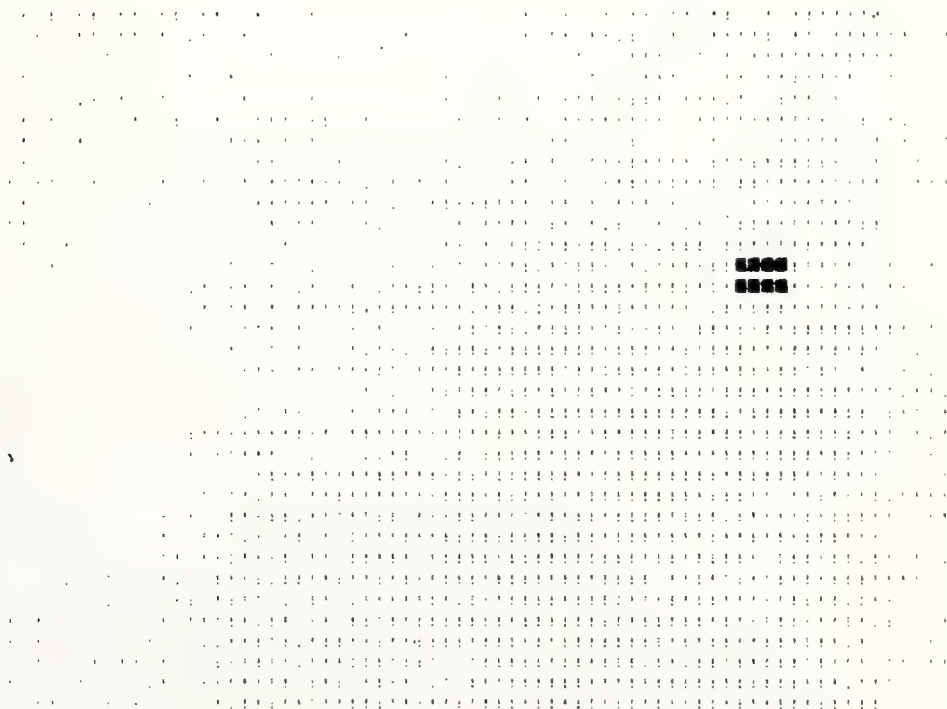


Figure 8.5 False alarms of the recursive filter at frame 10, using two thresholds.

The results for the hybrid filter are shown in figures 8.6 through 8.9. Since this filter does not use recursion in time, its transient is only in space. Similar improvement was achieved by using two thresholds, as can be seen in figures 8.7 and 8.9, compared against figures 8.6 and 8.8. Observe that frames 2 and 10 (figures 8.7 and 8.9) have almost the same number of false alarms, as should be expected, since there is no transient in time.

Comparing these results we see that the recursive filter presented better performance than the hybrid filter. This result is according to the theoretical and experimental values of tables 8.1 through 8.3, where the recursive filter has better performance for high background-to-noise ratios, as is the case (BNR = 30 dB).

3. Background Suppression

In what follows we present some examples of target enhancement (or background suppression). Several parameters can be varied to evaluate the estimator-detectors performance. In these examples we have used the same kind of background, but different values of target-to-background ratios.

The characteristics of the background are

$$\sigma^2 = 1, \quad \rho_v = \rho_h = 0.93, \quad \rho_t = 0.95, \quad \text{BNR} = 30 \text{ dB}$$

The correlations in space (ρ_v, ρ_h) have the same order of magnitude as those observed in chapter II for several real-life pictures. The background-to-noise ratio (BNR) was



Figure 8.6 False alarms of the hybrid filter at frame 2, using one threshold.

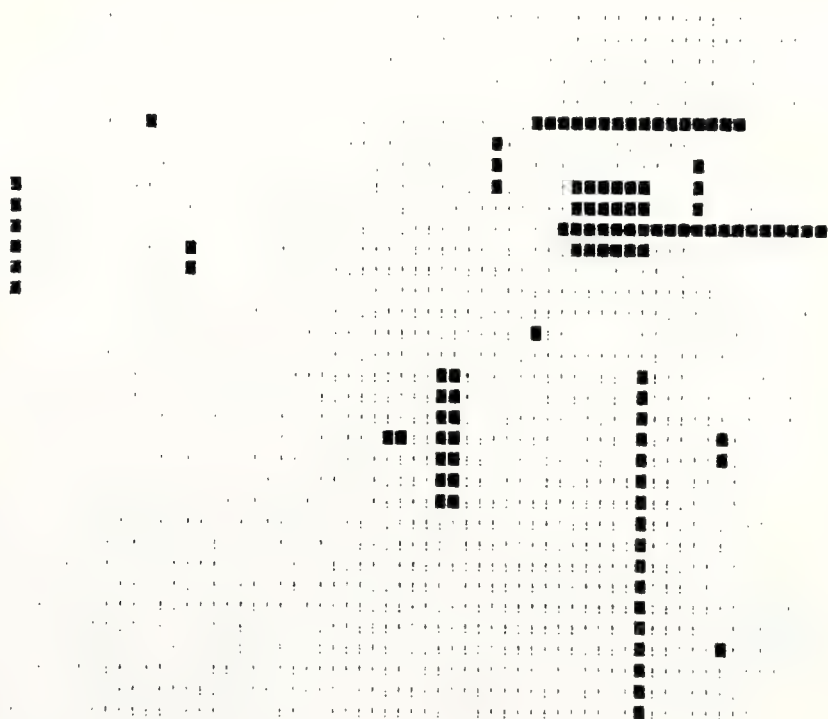


Figure 8.7 False alarms of the hybrid filter at frame 2, using two thresholds.

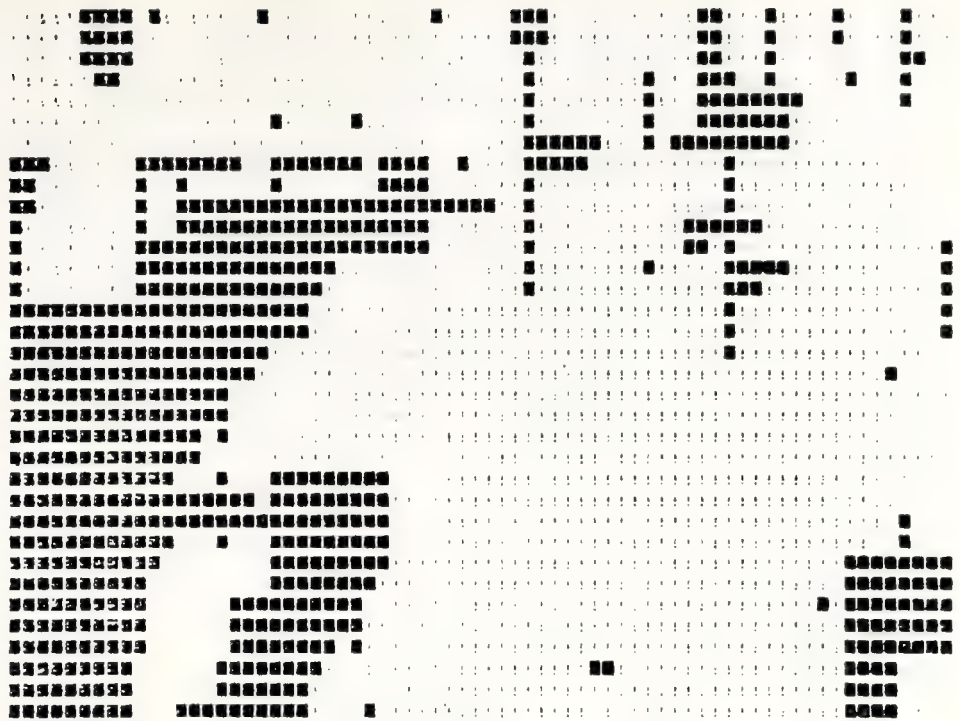


Figure 8.8 False alarms of the hybrid filter at frame 10, using one threshold.

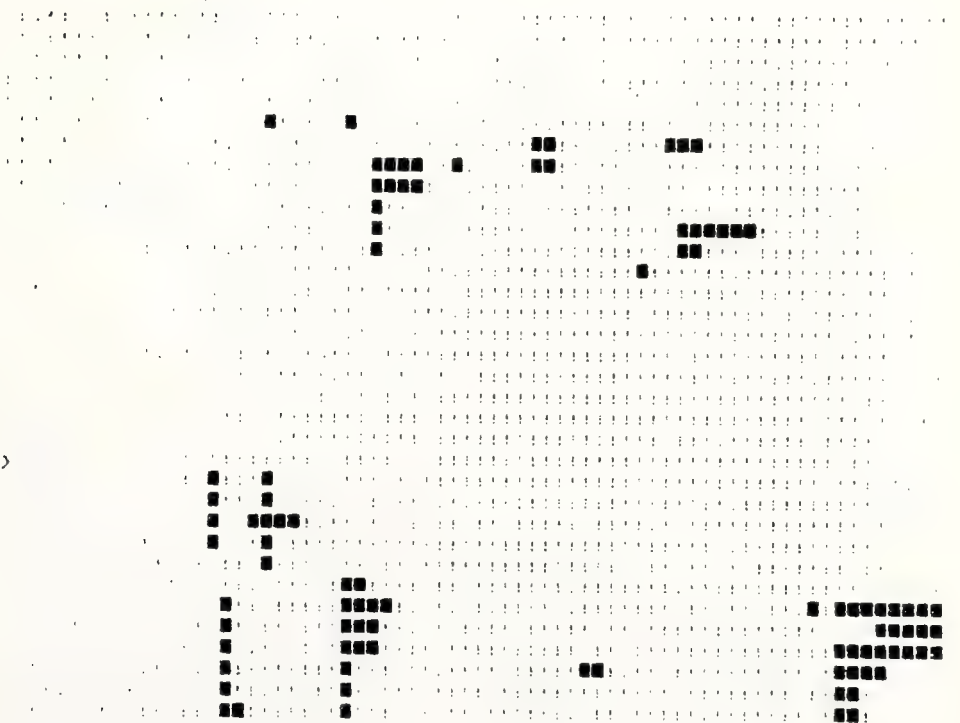


Figure 8.9 False alarms of the hybrid filter at frame 10, using two thresholds.

chosen so high, because we are interested here in those situations where the observation noise is neglectable when compared with the background, which is the undesirable texture to be suppressed, in order to enhance the target.

The performance of the filters will be given by the ratios defined in equations (8.1) through (8.3).

Example 1

Filter: Recursive

$TBR_i = 0 \text{ dB}$

Threshold = 0.5

Figures: 8.10 through 8.15

In table 8.4 the performance of the filter at several frames is shown. Observe that at frame 1 the 3D-recursive filter reduces to a 2D-recursive filter which does not use the correlation in time. Comparing figure 8.11 with 8.13 and 8.15 we can see the great improvement resulting from the exploitation of the correlation in time.

Example 2

Filter: Hybrid

$TBR_i = 0 \text{ dB}$

Threshold = 0.5

Figures: 8.16 and 8.17



Figure 8.10 Original image at frame 1 ($TBR_i = 0$ dB)



Figure 8.11 Frame 1 after the recursive filter
($TBR_0 = 12$ dB)



Figure 8.12 Original image at frame 6 ($TBR_i = 0$ dB)

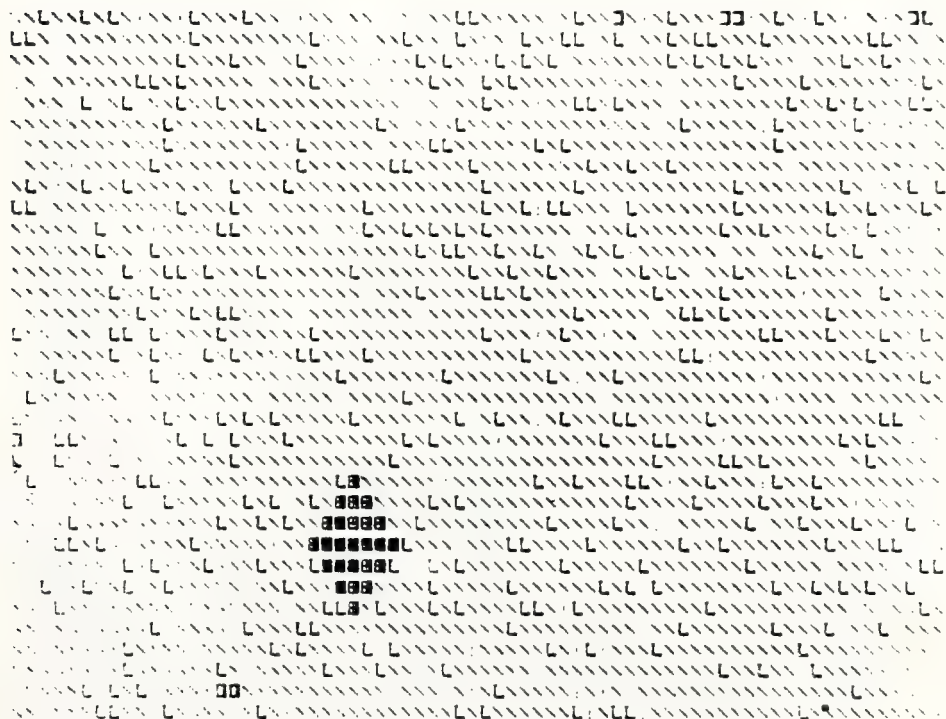


Figure 8.13 Frame 6 after the recursive filter
($TBR_0 = 17.3$ dB)



Figure 8.14 Original image at frame 10 ($TBR_1 = 0$ dB)

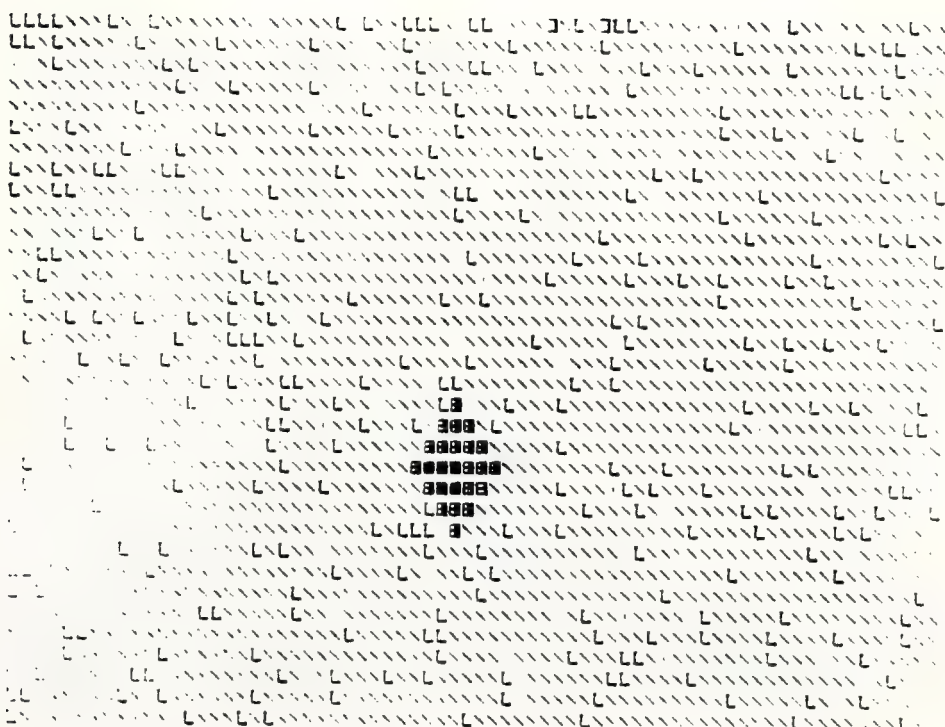


Figure 8.15 Frame 10 after the recursive filter ($TBR_0 = 17.3$ dB)



Figure 8.16 Frame 6 after the hybrid filter
($TBR_0 = 15 \text{ dB}$)

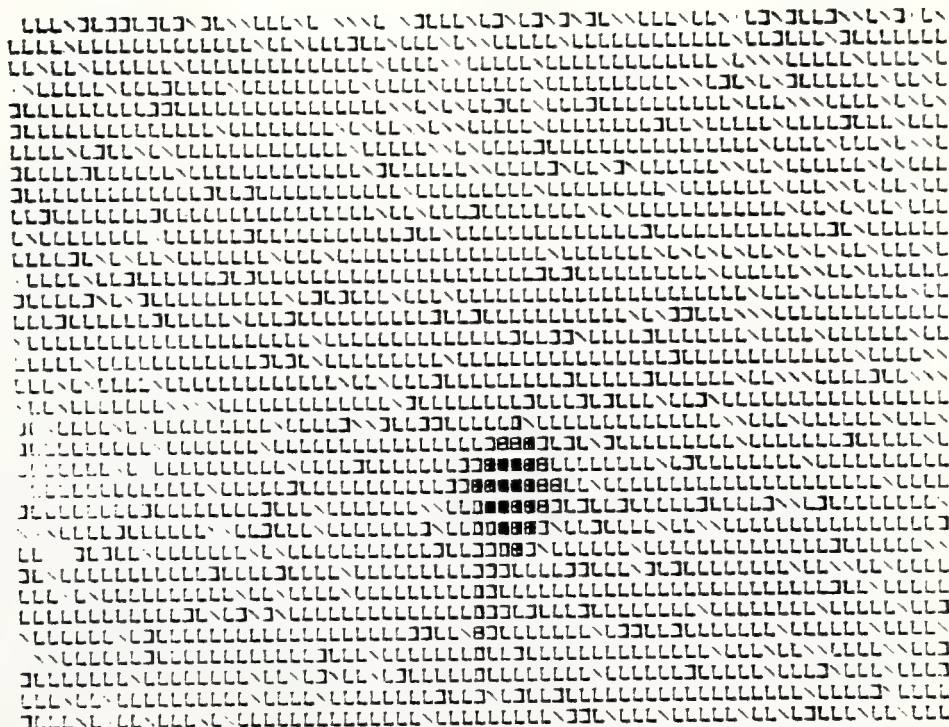


Figure 8.17 Frame 10 after the hybrid filter
($TBR_0 = 15 \text{ dB}$)

TABLE 8.1

3D-RECURSIVE FILTER

| FRAME | $TBR_i = 0$ dB (example 1) | | $TBR_i = -3$ dB (example 3) | |
|--|-------------------------------|-----------|--------------------------------|-----------|
| | TBR_0 (dB) | P.G. (dB) | TBR_0 (dB) | P.G. (dB) |
| 2 | 17.3 | 17.3 | 13.0 | 16.0 |
| 3 | 17.3 | 17.3 | 13.6 | 16.6 |
| 4 | 17.4 | 17.4 | 14.8 | 17.8 |
| 5 | 17.0 | 17.0 | 15.3 | 18.3 |
| 7 | 17.2 | 17.2 | 15.8 | 18.8 |
| 9 | 17.7 | 17.7 | 16.3 | 19.3 |
| 13 | 17.9 | 17.9 | 16.6 | 19.6 |
| 17 | 17.9 | 17.9 | 16.4 | 19.4 |
| 19 | 17.2 | 17.2 | 16.0 | 19.0 |
| 20 | 18.8 | 18.8 | 17.3 | 20.3 |
| AVERAGE OVER FRAMES 6 THROUGH 20 | 17.7 | 17.7 | 16.3 | 19.3 |

This example is similar to the previous one, but the hybrid instead of the recursive filter is used. The frame 1 is not shown, since both filters reduce to a 2D-recursive filter for this frame. Comparing the results we can see that the recursive filter (example 1) presented better performance than the hybrid one. As expected, the hybrid filter had almost constant performance (15 dB of processing gain) for all frames, since it has no transient in time.

Example 3

Filter: Recursive

$TBR_i = -3 \text{ dB}$

Threshold = 0.35

Figures: 8.18 through 8.23

The filter performance for other frames is shown in table 8.4, where the effect of the transition in time at the first 5 frames is clearly shown.

Example 4

Filter: Hybrid

$TBR_i = -3 \text{ dB}$

Threshold = 0.4

Figures: 8.24 and 8.25

This example is similar to the previous one, but using the hybrid instead of the recursive filter. The processing gain was around 12.5 dB for all frames. From figures 8.24 and 8.25 we can see that $TBR_i = -3 \text{ dB}$ is about the limit of detection for the hybrid filter.

4. Target Tracking

In what follows we present some examples of target detection and tracking from frame to frame. The background has the same characteristics as before. The first five frames are entirely processed in order to acquire the target



Figure 8.18 Original image at frame 1 ($TBR_i = -3$ dB)

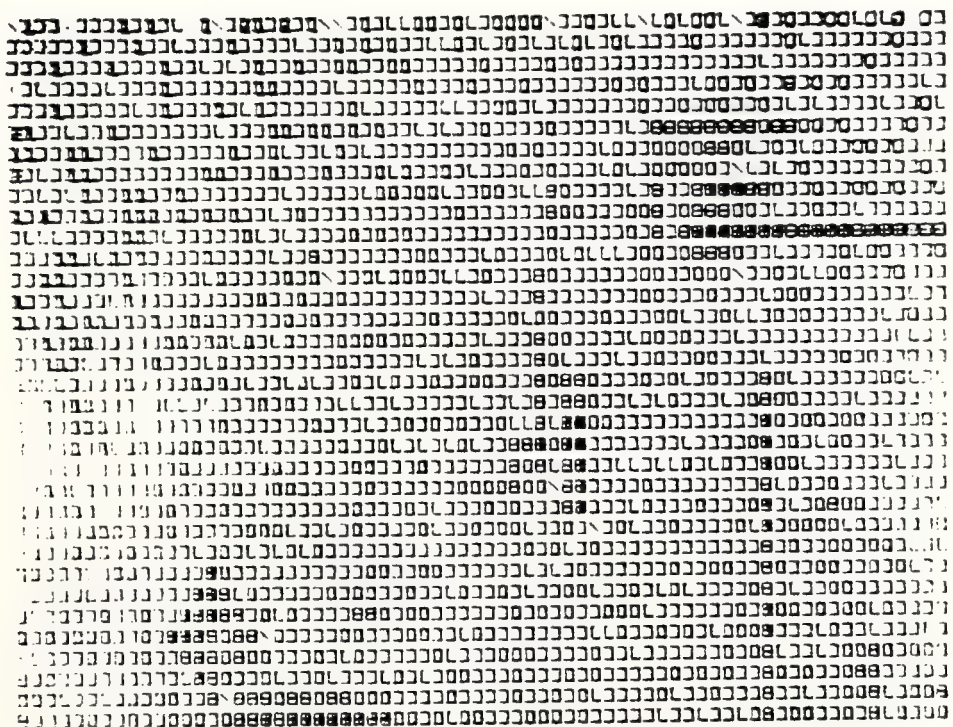


Figure 8.19 Frame 1 after the recursive filter
($TBR_0 = 9$ dB)

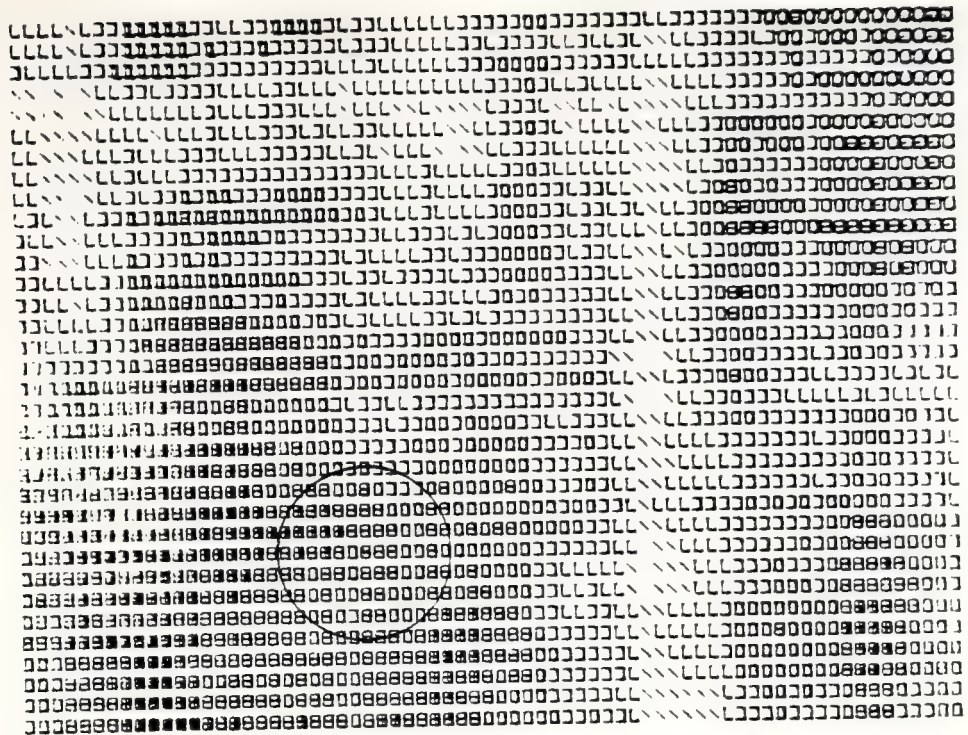


Figure 8.20 Original image at frame 6 ($TBR_i = -3$ dB)

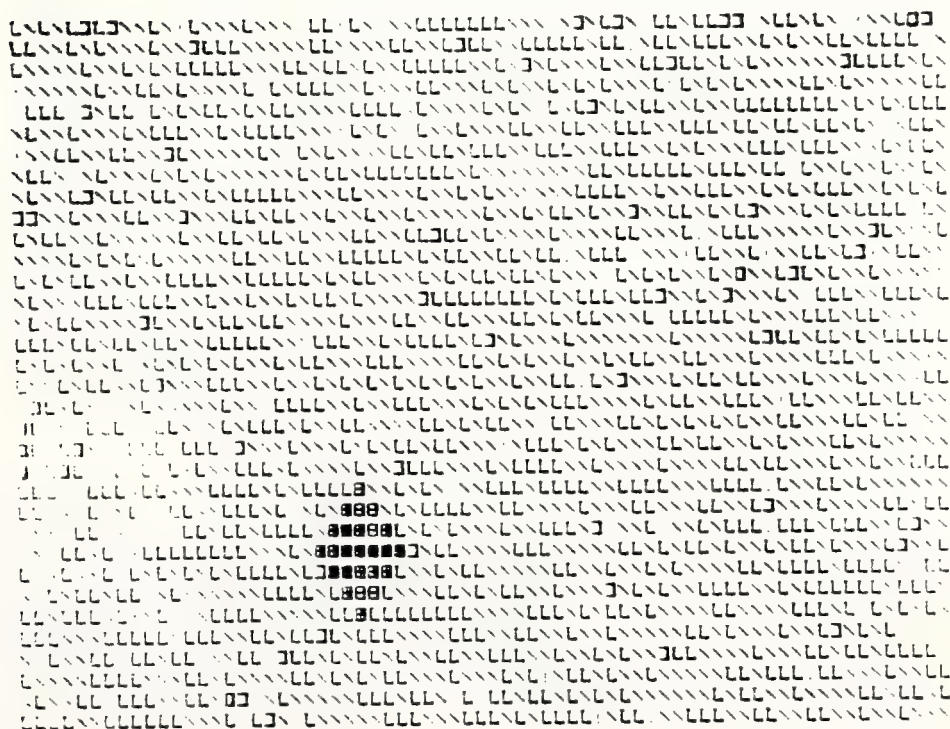


Figure 8.21 Frame 6 after the recursive filter
($TBR_0 = 15.8$ dB)



Figure 8.22 Original image at frame 10
($TBR_i = -3$ dB)

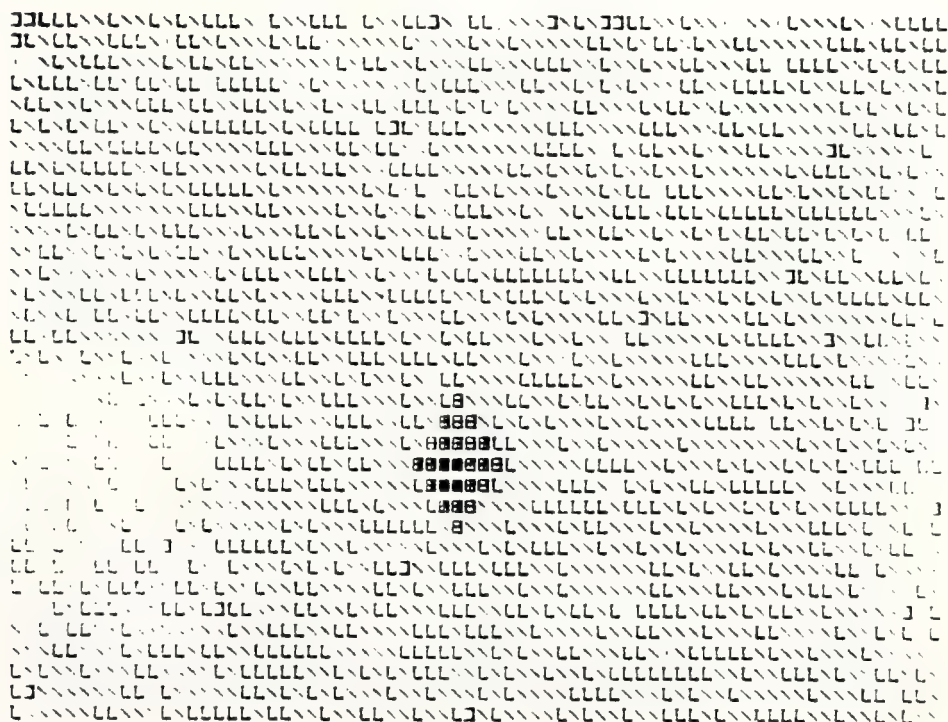


Figure 8.23 Frame 10 after the recursive filter
($TBR_0 = 15.9$ dB)

(search mode). The window mode is implemented at frame 6 and, thereafter, the pictures are processed only in a small 20 x 20 window centered at the target predicted position. In the examples we have created binary pictures to emphasize the performance of the filters.

Example 5

Filter: Recursive

$TBR_i = 0$ dB

Threshold = 0.5

Figures: 8.26 through 8.32

Example 6

Filter: Hybrid

$TBR_i = 0$ dB

Threshold = 0.5

Figures 8.33 through 8.36

Observe that the performance of the hybrid filter is inferior to that of the recursive filter in example 5. It can also be seen, in figure 8.36, that the tracking is better after the first 5 frames. The reason is that in the window mode the measurement error of the centroid position is limited by the size of the window.



Figure 8.28 Original image at frame 12
($TBR_1 = 0$ dB)

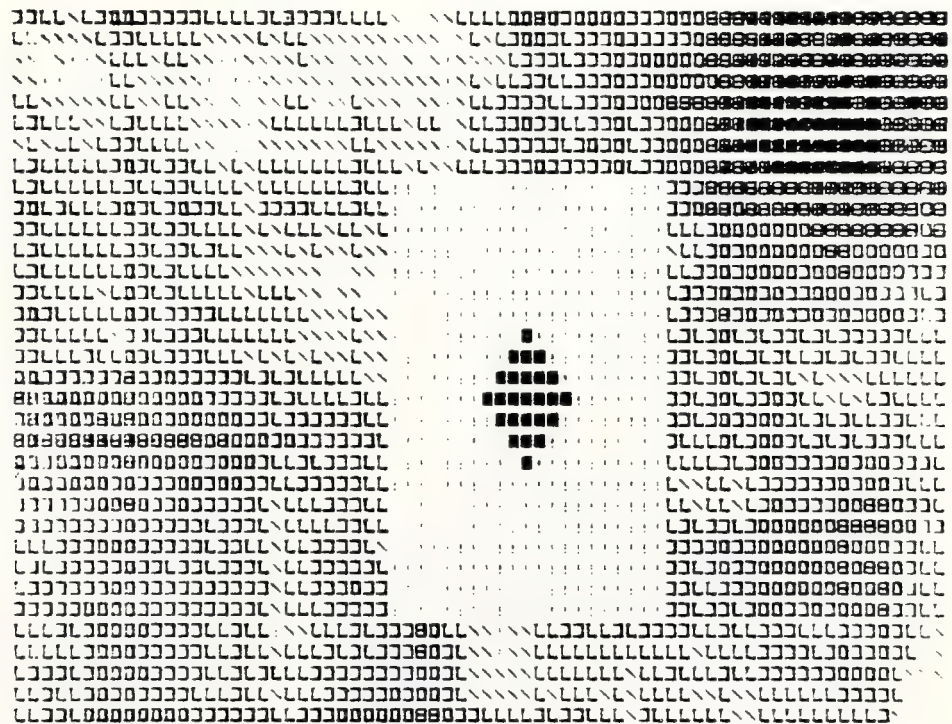


Figure 8.29 Frame 12 after the recursive filter



Figure 8.30 Original image at frame 18
($TBR_i = 0$ dB)

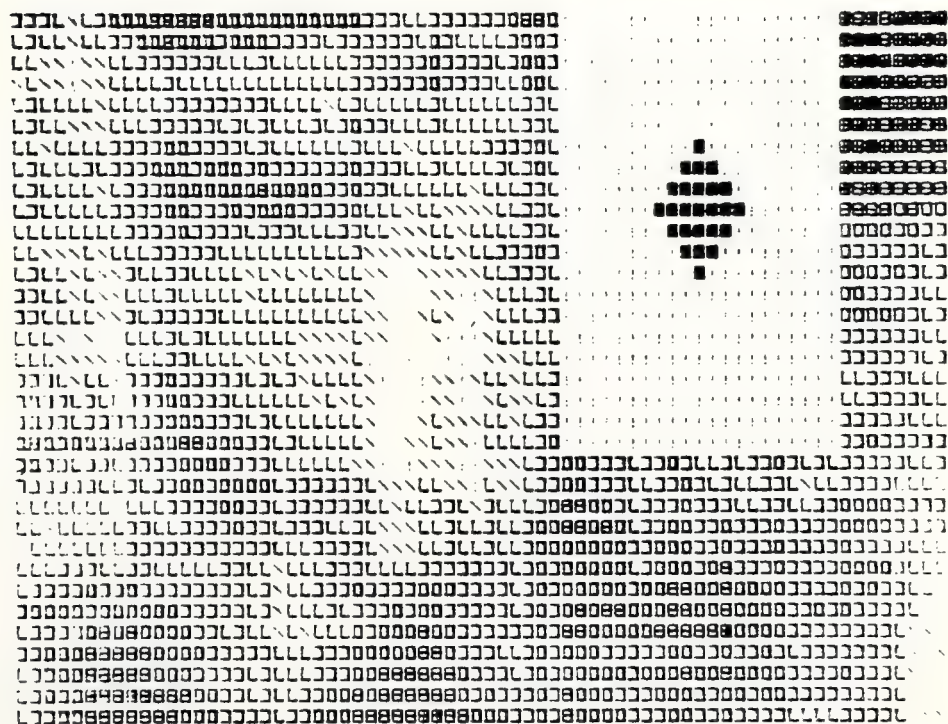


Figure 8.31 Frame 18 after the recursive filter

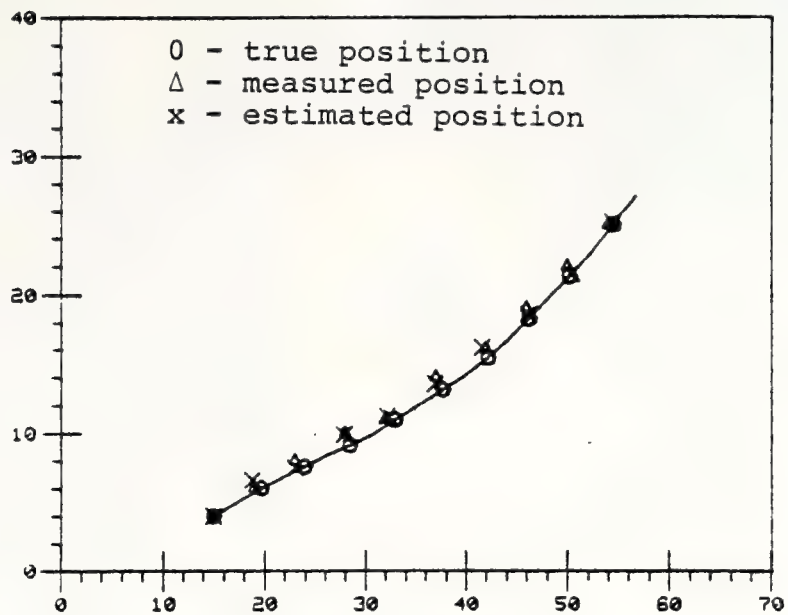


Figure 8.32 Target tracking with the recursive filter (example 5)



Figure 8.33 Frame 6 after the hybrid filter

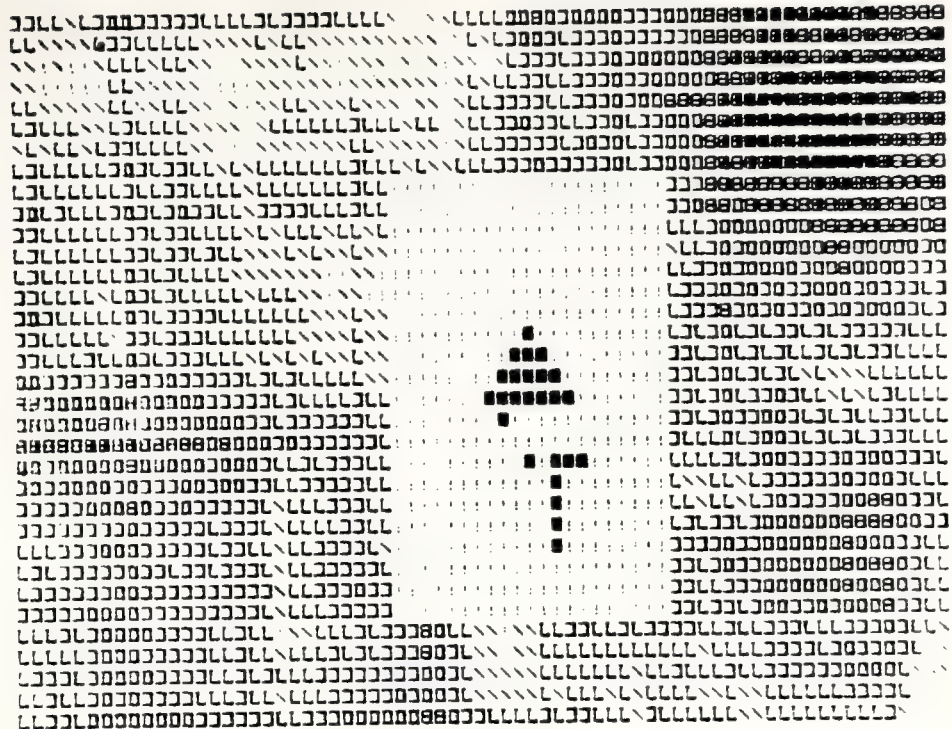


Figure 8.34 Frame 12 after the hybrid filter

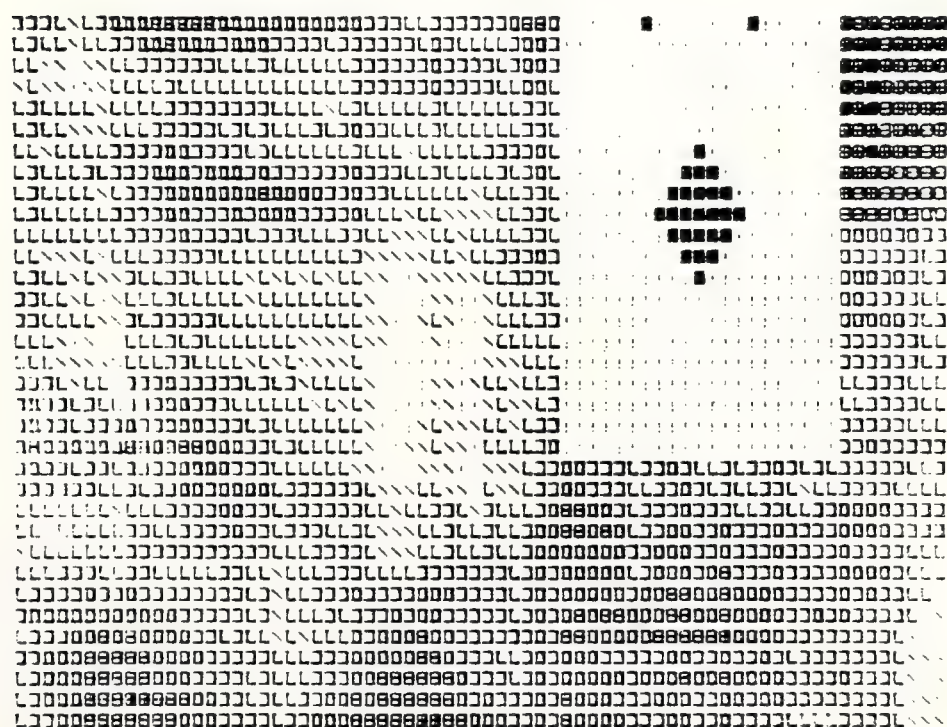


Figure 8.35 Frame 18 after the hybrid filter

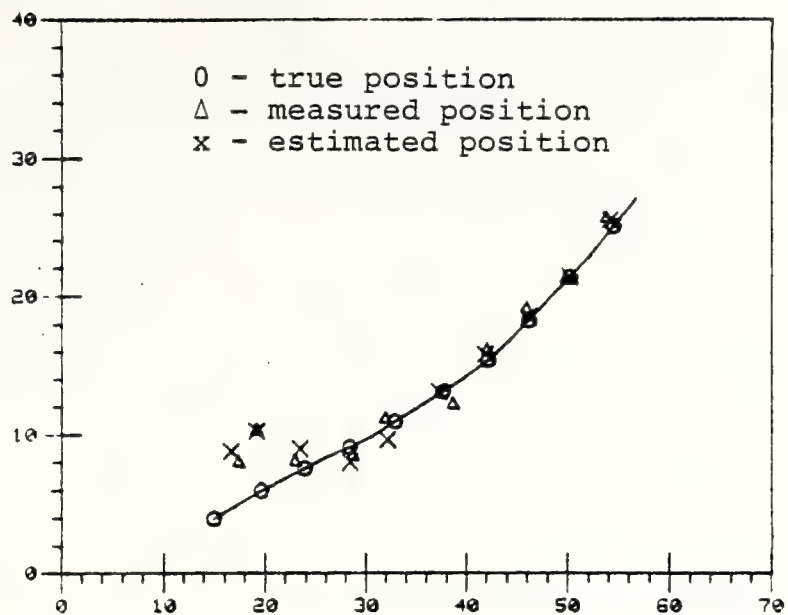


Figure 8.36 Target tracking with the hybrid filter (example 6)

Example 7

Filter: Recursive

$TBR_i = -3 \text{ dB}$

Threshold = 0.4

Figures: 8.37 through 8.43

Example 8

Filter: Hybrid

$TBR_i = -3 \text{ dB}$

Threshold = 0.4

Figures: 8.44 through 8.47

Comparing examples 7 and 8 we see that both filters were able to track the target, but the recursive filter is clearly superior to the hybrid one.



Figure 8.37 Original image at frame 6 ($TBR_1 = -3$ dB)



Figure 8.38 Frame 6 after the recursive filter

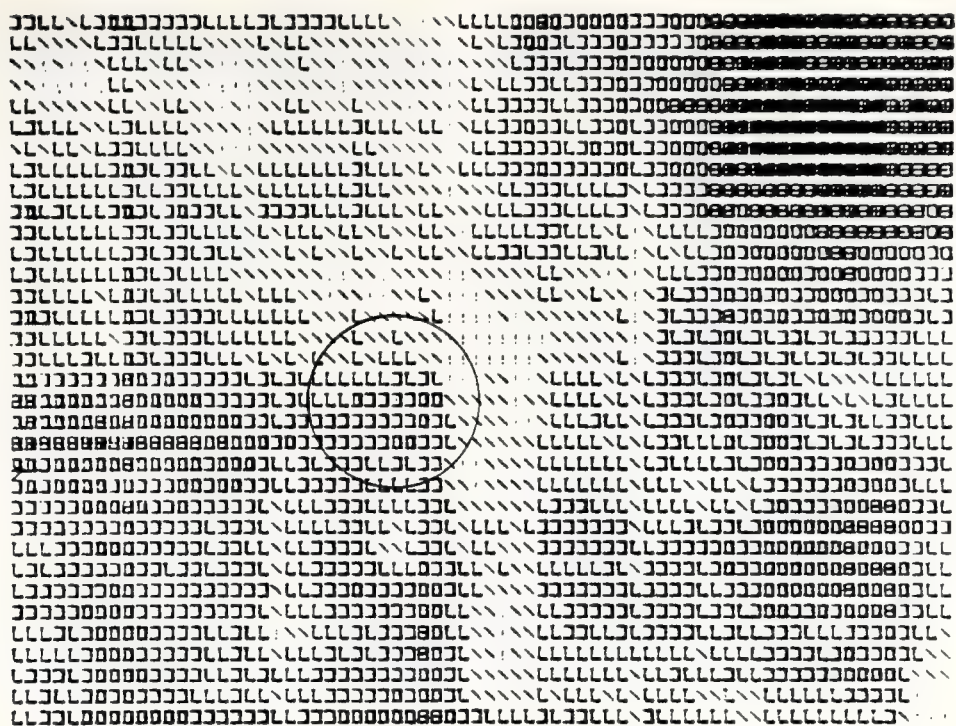


Figure 8.39 Original image at frame 12 ($TBR_i = -3$ dB)



Figure 8.40 Frame 12 after the recursive filter

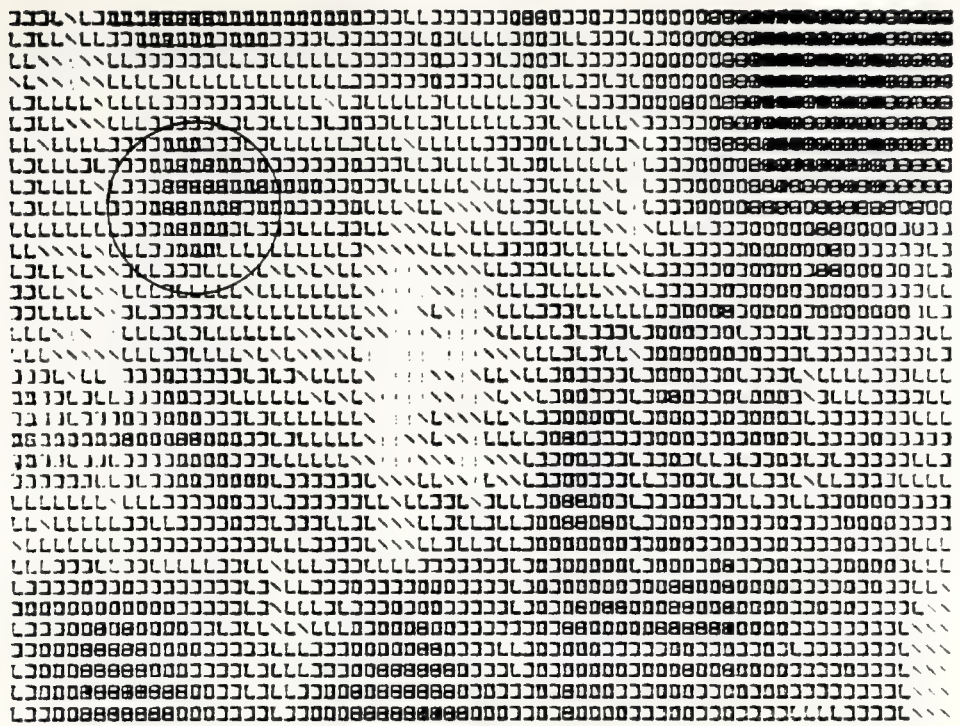


Figure 8.41 Original image at frame 18 ($TBR_i = -3$ dB)

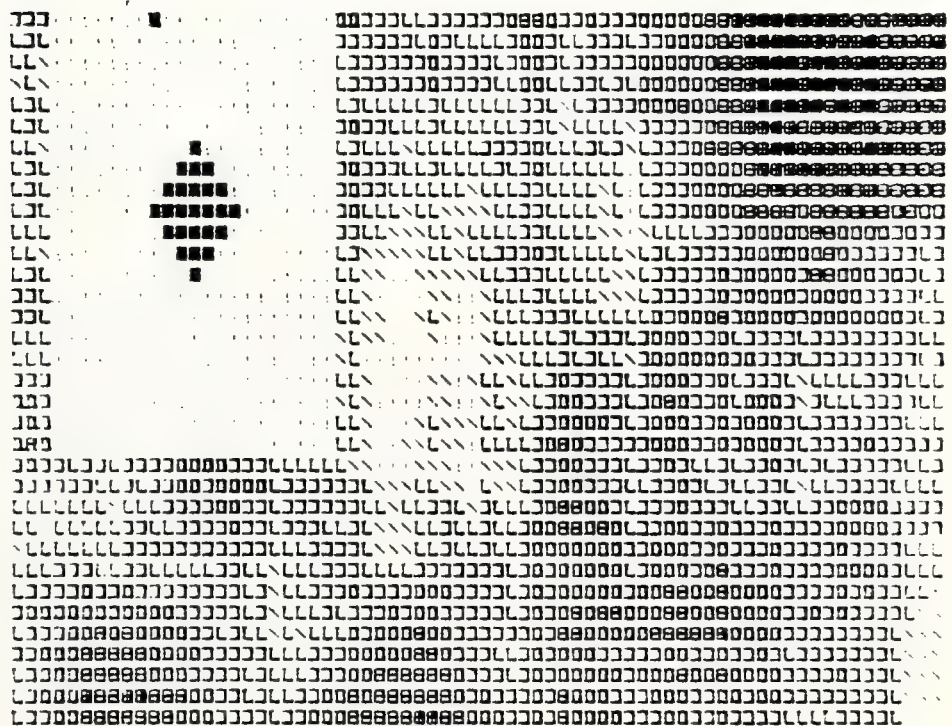


Figure 8.42 Frame 18 after the recursive filter

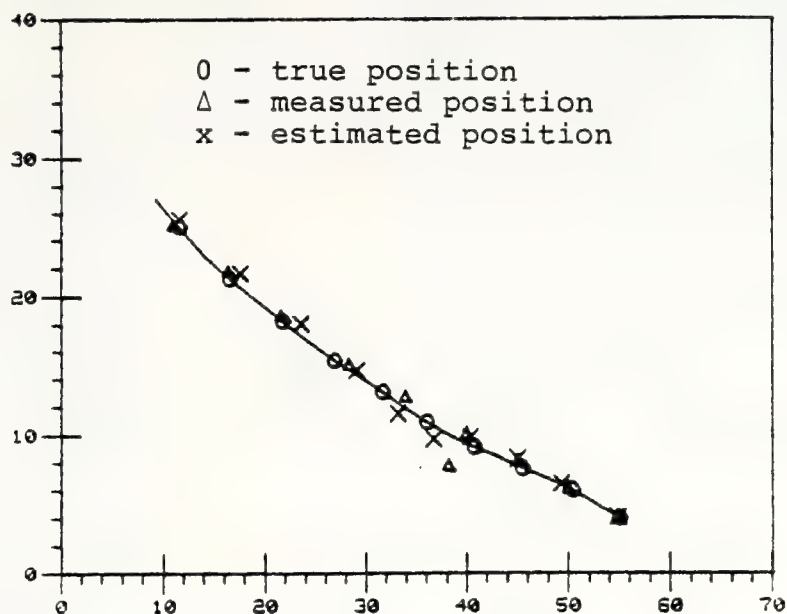


Figure 8.43 Target tracking with the recursive filter (example 7)

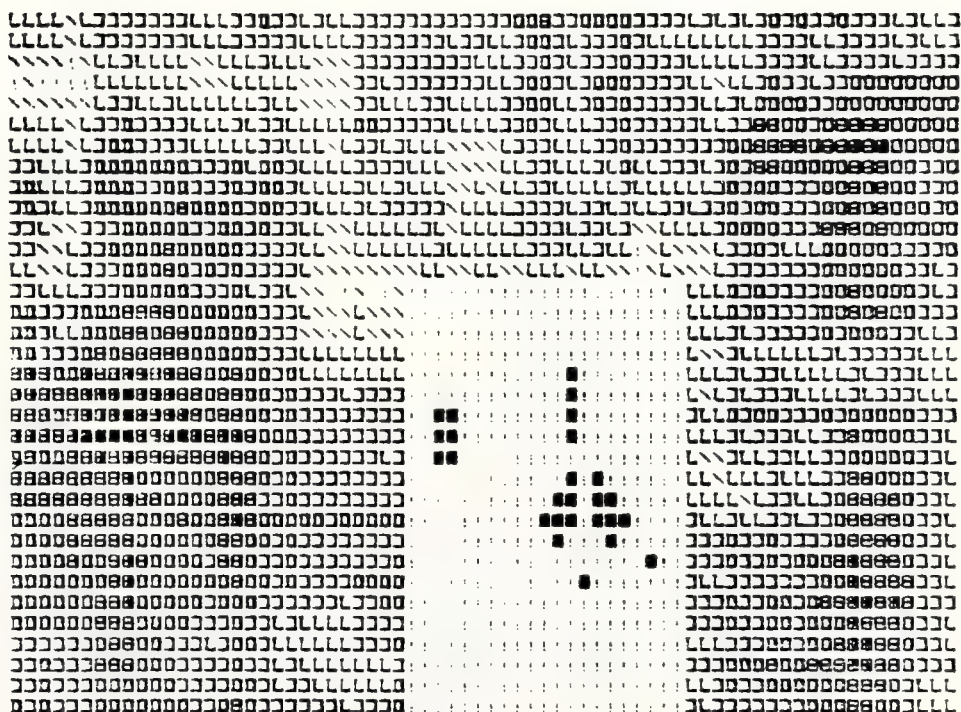


Figure 8.44 Frame 6 after the hybrid filter

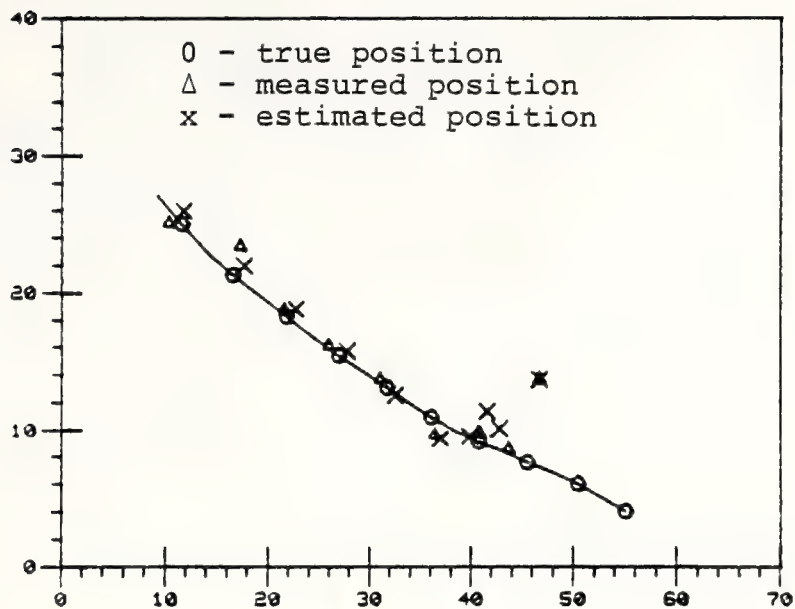


Figure 8.47 Target tracking with the hybrid filter (example 8)

IX. CONCLUSIONS

A. SUMMARY OF RESULTS

The experiments with several real life pictures allow us to draw two important conclusions concerning image modeling. First, the hypothesis of an autocorrelation function with separable kernels in the vertical and horizontal directions worked quite well. This conclusion is very important, because the derivation of the dynamic model is substantially simplified by avoiding the difficult problem of two-dimensional spectral factorization. Second, the first-order model, used by most researchers, is a poor approximation for pictures with few details, although it is a very good model for pictures with many details. The autocorrelation function, suggested in this research, seems to be a good choice, because it includes the first and second order models, as particular cases, and also permits a simple method of parameter identification which can be easily implemented.

The performance evaluation of the sub-optimum two-dimensional recursive filters of Habibi's type [3-5] allow us to draw some interesting conclusions. Although sub-optimum, these filters are not so far from optimality. In the worst case, the error variance was around 15% greater than the one obtained by the optimum interpolator. Rosenfeld's filter [3] is the best and presented degradation

around 8%. Shachar's filter [5] has the simplest algorithm for gain calculation which uses a very good approximation, since its performance was quite close to that of a similar filter suggested here which calculates the gains without approximation.

The reason why Rosenfeld's filter [3] is the best is because the estimation error is orthogonal to the observation of the pixel under estimation, although this does not happen for the rest of the data set. This conclusion leads directly to the hybrid filters introduced in this research. From the results shown in Table 5.1, it can be seen that the non-recursive filter [6], suggested by Bar-Yehoshua [6], although using only 9 observations, presents better performance than the optimum (in the sense of recursive filtering) interpolator, for correlations as high as 0.94. The conclusion is that most of the information about the pixel gray level under estimation resides on its closest neighbors. For this reason, the hybrid filter presents the best performance, because it makes use of all the observations in the neighborhood and also those farther observations used by the recursive filter. Experiments with real life pictures (figures 5.12 and 5.13) show the ability of the hybrid filter in smoothing out observation noise in pictures with signal-to-noise ratio as low as -3 dB. The processing gain for this case was 11.1 dB, which compares favorably with previous results.

For images sequenced in time, the three-dimensional recursive and hybrid filters were developed in order to exploit the correlation in time. The recursive filter is adequate for the cases where the scene is the same during several frames, otherwise the exploitation of the time correlation is not advantageous, due to the filter transient at the first few frames. Its structure is intuitively appealing, since it reduces to the one-dimensional Kalman filter for the case of no spatial correlation, and to a two-dimensional recursive filter for the case of no correlation in time. Numerical results show that the use of time correlation can be quite advantageous for the case of time correlation greater than spatial correlation (error variance is reduced by 33%), but it is not so advantageous for the opposite case (error variance is reduced by 6%). The hybrid filter is adequate for the case where the scene changes at every two frames, since it makes use of only the previous and the present frames. Of course, it can also be designed for the case of a fixed scenario (see Chapter V) in which case it improves the performance of the recursive filter.

To detect targets from cluttered background images a likelihood ratio was constructed which uses the observation of the pixel and the background prediction for the pixel gray level. A quite general situation was considered. The target and background are considered as two kinds of textures. The image is modeled as a weighted combination of

three additive components: target, background and observation noise. Therefore, it can accommodate the case of infrared pictures and also visible pictures, where there is a replacement process, target or background. The prediction of the background is given by the recursive and hybrid filter or, also, by a non-recursive filter. The likelihood ratio was particularized for four special cases of interest, which had their performance analyzed. The likelihood ratio enhanced the target-to-background ratio. The target is extracted by threshold detection. The threshold is chosen in order to maintain the probability of false alarm below some desired value. The performance of the detection process is highly related to the background-to-observation noise ratio, since this ratio is responsible for the background prediction.

The target detection and tracking problem was simulated using computer generated images with characteristics similar to those of real life images used in the other experiments. Both three-dimensional recursive and hybrid predictors were used and compared. The likelihood ratio, using both filters, was able to detect a target at target-to-background ratios as low as -3 dB. The processing gain was around 19 dB, for the recursive filter, and around 12.5 dB for the hybrid filter. The recursive filter presented better performance, as should be expected from the results of Tables 5.3 through 5.5, where its prediction is better than that of the hybrid filter, for the case of high background-to-noise ratio.

B. SUGGESTIONS FOR FUTURE RESEARCH

Many pictures are clearly non-homogeneous, therefore the development of space-varying models is the natural way to proceed in order to improve the model suggested in this research. The recursive techniques are directly amenable to the analysis of space-varying models. The reliance on "a priori" information is a very important area. The method of parameter identification suggested here is applicable for the case of constant parameters for the whole frame. In the non-homogeneous case such identification has to be accomplished pixel by pixel, in order to improve the robustness of this statistical approach in face of modeling errors.

Due to the enormous computational load of the optimum multi-dimensional recursive filters, and also because the observations far away carry negligible information about the pixel to be estimated, the natural way is to look for sub-optimal recursive filters that require less computation.

The hybrid filters suggested here can be equally applied with other recursive filters, as, for example, the reduced dimension filters proposed by Panda and Kak [9] and Woods and Radiwan [11].

The decision rule suggested here is directed to the detection of targets pixel by pixel, in order to be independent of the target shape. The choice of the threshold was based on the probability of a false alarm of each isolated

decision. Since a wrong decision necessarily affects the next, a question arises concerning how to vary the threshold in order to keep the false alarm probability below the desired value.

LIST OF REFERENCES

1. E.R. Kretzmer, "Statistics of Television Signals," Bell System Technical Journal, July 1952, pp. 751-763.
2. Habibi, A., "Image Coding by Linear Transformation and Block Quantization," Trans. IEEE, Vol. Com-19, No. 1, February 1971, pp. 50-52.
3. A. Rosenfeld, and A.C. Kak, Digital Picture Processing, Academic Press, New York, 1976.
4. Habibi, A., "Two-Dimensional Bayesian Estimation of Images," Proc. IEEE, Vol. 60, No. 7, July 1972, pp. 878-883.
5. Shachar, Moshe, "Modeling and Recursive Estimation of Two-Dimensional Random Fields and Applications to Target Detection," EE Thesis, Naval Postgraduate School, Monterey, California, June 1977.
6. Bar-Yehoshua, David, "Two-Dimension Non-Recursive Filter for Estimation and Detection of Targets," EE Thesis, Naval Postgraduate School, Monterey, California, June 1977.
7. Nahi, N.E., and C.A. Franco, "Recursive Image Enhancement-Vector Processing," Proc. IEEE, Vol. C-24, April 1973, pp. 305-310.
8. Nahi, N.E., "Role of Recursive Estimation in Statistical Image Enhancement," Proc. IEEE, Vol. 60, No. 7, July 1972, pp. 872-878.
9. Panda, D.P. and A.C. Kak, "Recursive Least Squares Smoothing of Noise in Images," Trans. IEEE, Vol. ASSP-25 No. 6, December 1977, pp. 520-524.
10. Strintzis, M.G., "Comments on Two-Dimensional Bayesian Estimate of Images," Proc. IEEE, Vol. 64, p. 1255, August 1976.
11. Woods, J.W. and C.H. Radewan, "Kalman Filtering in Two-Dimensions," IEEE Trans. Inform. Theory, Vol. IT-23, No. 4, pp. 473-482, July 1977.
12. Willsky, A.S., "Relationship Between Digital Signal Processing and Control and Estimation Theory," Proc. IEEE, Vol. 66, No. 9, pp. 996-1017, September 1978.

13. Ekstrom, M.P. and J.W. Woods, "Two-Dimensional Spectral Factorization with Applications in Recursive Digital Filtering," Trans. IEEE, Vol. ASSP-24, No. 2, pp. 115-128, April 1976.
14. Jain, A.K. and E. Angel, "Image Restoration Modeling, and Reduction of Dimensionality," IEEE Trans. on Comput., Vol. C-23, pp. 470-476, May 1974.
15. Van Trees, H.L., Detection, Estimation, and Modulation Theory, Part I, John Wiley and Sons, Inc, N.Y., 1968.
16. Gelb, A., Applied Optimal Estimation, The M.I.T. Press, Cambridge, Massachusetts, 1974.
17. Kwakernaak, H. and R. Sivan, Linear Optimal Control Systems, Wiley-Interscience, a Division of John Wiley & Sons, Inc., New York, 1972.
18. Fukunaga, K. et al., "Segmentation and Structure Analysis for Real-time Video Tracking System," Technical Report No. STEWS-ID-77-1, October 1977, U.S. Army White Sands Missile Range, New Mexico.
19. Flachs, G.M., et al., "A Pre-prototype Real-time Video Tracking System," Final report for contract DAAD07-76-C0024, January 1977, Department of Electrical and Computer Engineering, New Mexico State University.
20. Oppenheim, A.V. and R.W. Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

INITIAL DISTRIBUTION LIST

| | No Copies |
|--|-----------|
| 1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314 | 2 |
| 2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940 | 2 |
| 3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940 | 1 |
| 4. Professor H.A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940 | 10 |
| 5. Professor T.F. Tao, Code 62TV Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940 | 1 |
| 6. Professor J.P. Powers, Code 62Po Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940 | 1 |
| 7. LCDR José Leite Pereira F ^a Rua Sá Ferreira, 128 Apt. 801 Copacabana - Rio de Janeiro RJ-20000 BRASIL | 3 |
| 8. Diretoria de Armamento e Comunicacoes Ministerio da Marinha Rua Visconde de Inhauma - Rio de Janeiro RJ-20000 BRASIL | 1 |
| 9. Instituto de Pesquisas da Marinha Rua Ipiru S/N Ilha do Governador - Rio de Janeiro RJ-20000 BRASIL | 1 |

Thesis
P3347
c.1

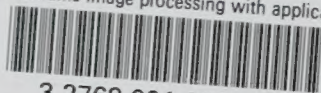
180835
Pereira Filho
Interframe image
processing with
application to
target detection
and tracking.

Thesis
P3347
c.1

Pereira Filho 180835
Interframe image
processing with
application to
target detection
and tracking.

thesP3347

Interframe image processing with applica



3 2768 001 00219 9

DUDLEY KNOX LIBRARY